Fraser Stream Integration

Application Data Sheet

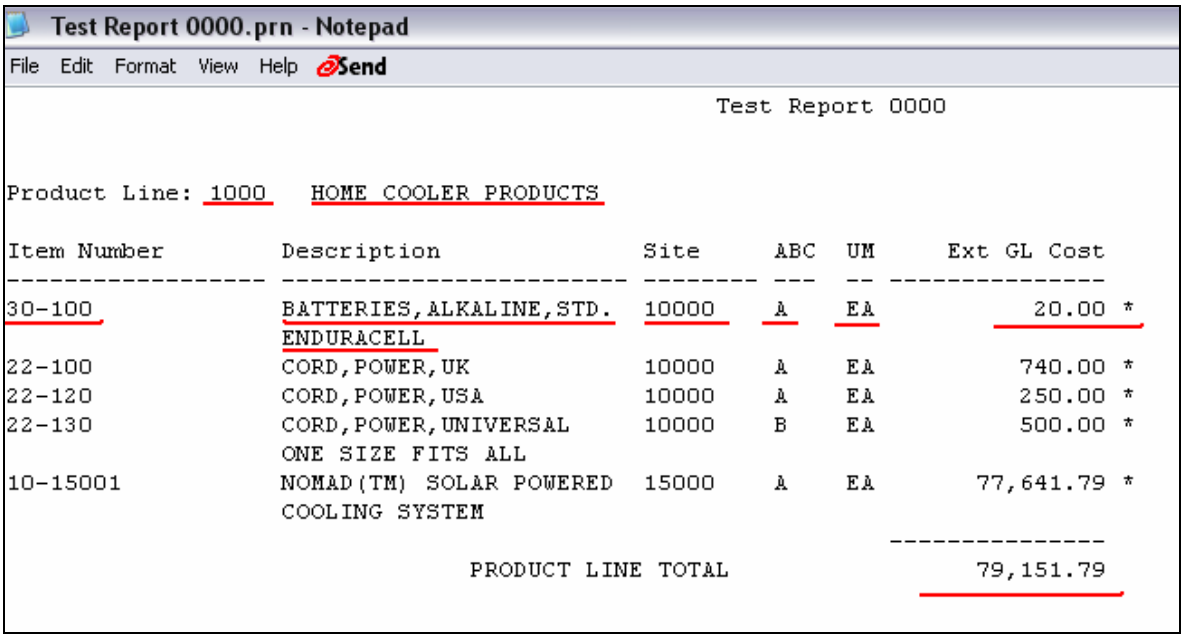
1. Fraser Stream Integration

Fraser Stream Integration (FSI) is a powerful data transformation tool that rapidly transforms structured ASCII text reports and delimited files into various file formats.

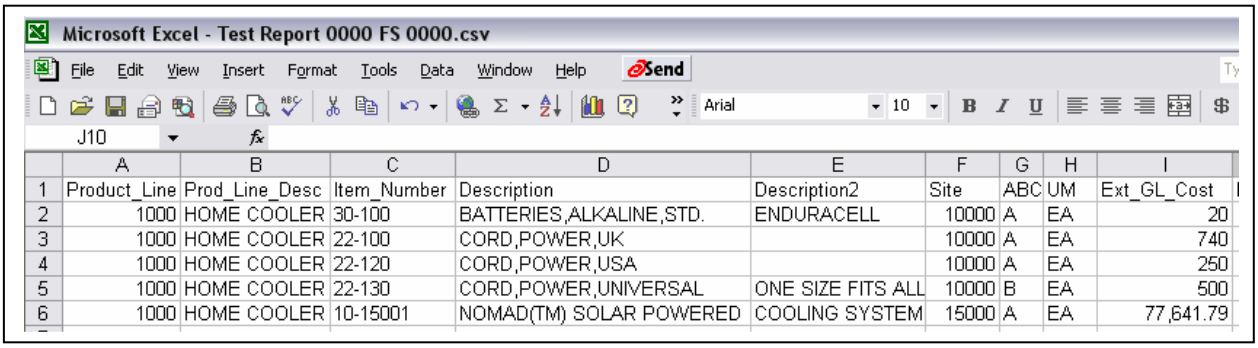
- Translate text reports from any source into:
  - Excel
  - Professional looking PDF
  - Word
  - XML with style sheets
  - Dataset
  - CSV
- Extract information from text reports and deposit directly into any database
- Publish information onto the web as HTML, XML with style sheets or PDF
- Output to XML format for integration with other applications or to share with business partners

Regardless of what the end use will be, the backbone of Fraser Stream technology is the method in which text reports are translated into the aforementioned formats. The *TRANSFORMATION SETTINGS* define how each type of report is to be processed and converted. Mapping transformation settings takes a matter of a few minutes and needs only to be done once. Once the settings are saved, the text report can be converted at the click of a button or scheduled to be processed at any predefined interval. Transformation Settings map information from the text reports header, detail, and footer sections, then translate this information into a single record which in turn can be displayed as a single row of an Excel spreadsheet or exported as unique record into a database.
Transformation Settings map information from the text report's header, detail, and footer sections, then translate this information into a single record which in turn can be displayed as a single row of an Excel spreadsheet or exported as unique record into a database.

The following example depicts a typical text report with the subsequent FSI output to Excel:



The same report in Excel format:

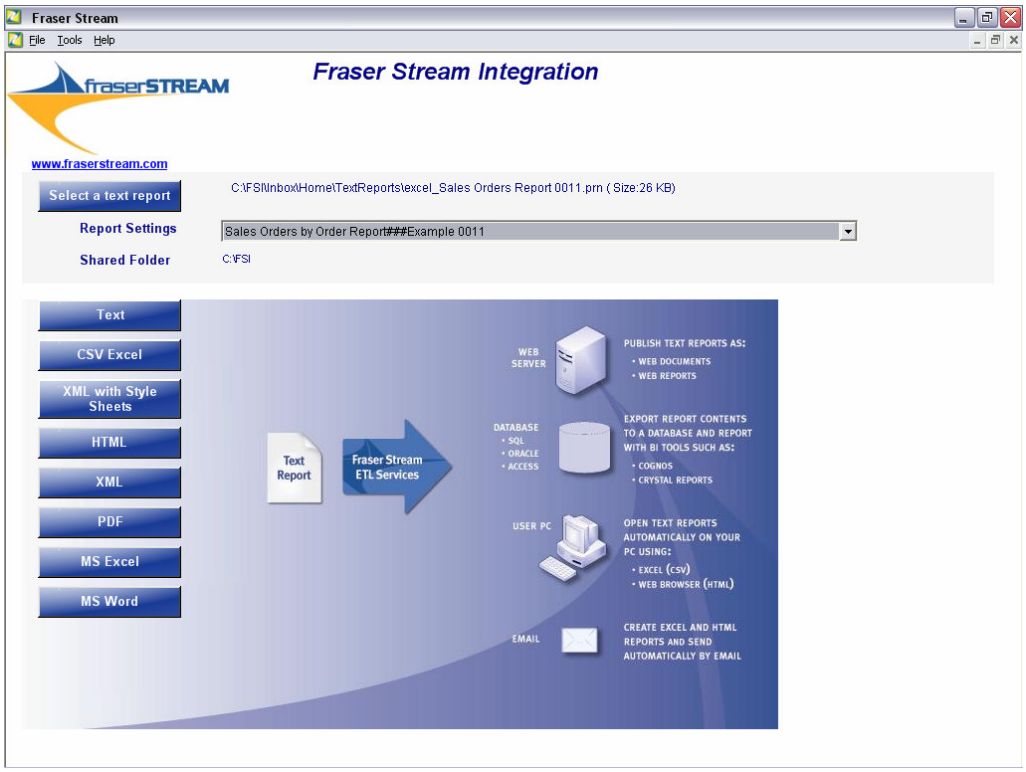


	A	B	C	D	E	F	G	H	I
1	Product_Line	Prod_Line_Desc	Item_Number	Description	Description2	Site	ABC	UM	Ext_GL_Cost
2	1000	HOME COOLER	30-100	BATTERIES,ALKALINE,STD.	ENDURACELL	10000	A	EA	20
3	1000	HOME COOLER	22-100	CORD,POWER,UK		10000	A	EA	740
4	1000	HOME COOLER	22-120	CORD,POWER,USA		10000	A	EA	250
5	1000	HOME COOLER	22-130	CORD,POWER,UNIVERSAL	ONE SIZE FITS ALL	10000	B	EA	500
6	1000	HOME COOLER	10-15001	NOMAD(TM) SOLAR POWERED	COOLING SYSTEM	15000	A	EA	77,641.79

Fraser Stream Integration is easy to use, requires very little technical know-how, and costs a fraction of what is currently offered on the market today. This Microsoft Windows based tool can be used as a stand alone desktop application or run on a server as an enterprise level ETL (extract load transform) solution.

1.1 Fraser Stream Output Types

Fraser Stream Integration (FSI) transforms ASCII text reports or delimited files into EXCEL, PDF, CSV, HTML or XML with Style Sheets.



1. Click on the **Select a Text Report** button
2. Select the folder: C:\Program Files\Fraser Stream\Fraser Stream Integration\TextReports
3. Select the text file: **Test Report 0001.prn**
   (Transformation settings will automatically appear in the drop down list)
4. Click on the **Text** button to view the report in its original format in Notepad

5. Click on the **CSV Excel** button to see the text report in Excel format. The application will extract report header, details line and footer information and display each record on a single Excel row
6. Click on the **HTML** button to see the text report in HTML format
7. Click on the **XML** button to see the text report in XML format
8. Click on the **XML with Style Sheets** button to open the XML file using a Style Sheet. You may customize the Style Sheet to display your company's logo, fonts, colors, background or any other HTML elements
9. The **PDF**, **MS Word** and **MS Excel** are created with Crystal Reports. Please see PDF Output chapter

Transformation Settings must be mapped in order for FSI to work. Fortunately this process only needs to be done once. Open FSI and click on the ***'Select a text report'*** button to choose the text file you want to transform. Fraser Stream Integration will scan the text report and check to see if transformation settings have been mapped for this report in the past.

NOTE: FSI does not rely on the report name to determine if transformation settings exist. Instead, FSI looks at the structure of the report and identifies the precise positions of certain key characteristics such as the header, detail, and footer sections to determine if this file has been mapped before.

If a specific type of report has already been mapped, a transformation setting will appear in the drop down list (see screen shot). A report can have multiple transformation settings. For example one set may include each detail column i.e. price, quantity etc…A second set of settings may only capture the header section and the sales order number. Fraser Stream Integration will display all previously mapped settings. Select the settings you wish to use from the drop down list. If no settings exist, you will need to create them. Please download the PDF file Fraser Stream Reports Mappings.

Assuming transformation settings exist, simply select the settings from the drop down list and click on any of the output buttons to view the data in the different formats.
Fraser Stream Software offers a free 30 day trial of Fraser Stream Integration. You may download this free version from the ***Downloads*** section of www.fraserstream.com.

Follow these steps to install the program:

1.2 Installation Requirements

1. Supported Operating Systems:

   Windows 2000, Windows 98, Windows ME, Windows NT, Windows Server 2003, Windows XP, Windows Vista
2. Processor: Minimum133-MHz Intel Pentium-class processor
3. Memory: Minimum128 MB of RAM, 256 MB recommended

1.3 Fraser Stream Installation

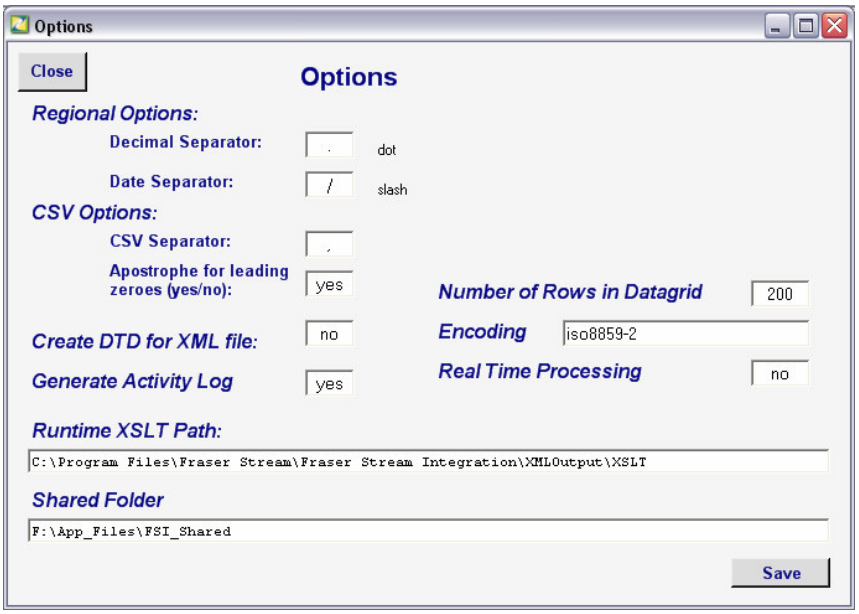
1. Go to: www.fraserstream.com/downloads
2. Click on Download Now For the software you want to install
3. Click Run to install the software
4. The software will be installed in the folder: ***.\Program Files\Fraser Stream\Fraser Stream Integration***
5. To Start Fraser Stream Integration: ***Start\All Programs\Fraser Stream Integration***

2 Fraser Stream Reports (Fraser Stream Integration Client)

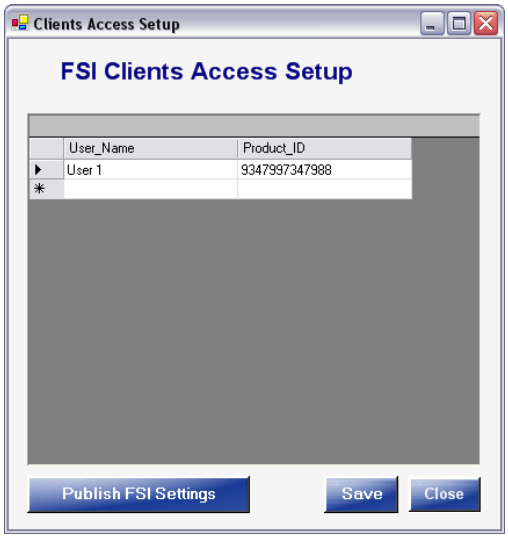
Fraser Stream Reports - Client has the same functionality as Fraser Stream Integration except batch functionality. Fraser Stream Reports - Client installation allows a company to create transformation settings for a large number of text reports that are shared by all company's users. Fraser Stream Reports – Client doesn't need a registration code but it has to be registered with Fraser Stream Integration (See step 3 below).

Fraser Stream Reports – Client must have access to the Shared Folder. (See step 1 below).

1. Create FSI Shared folder.
   - Create a network folder where Fraser Stream Reports users have access. This will be the Fraser Stream Shared Folder
   - Unzip "FSI_Shared.zip" file from "C:\Program Files\Fraser Stream\Fraser Stream Integration\Deployment" into the network folder
   - Update the Shared Folder path in "**Files\Options**" menu with "**[Network folder]\FSI_Shared** and click **Save** button



2. The transformation settings are created using Fraser Stream Integration and published into the Shared Folder on the network. Click "**Publish FSI Settings**" from "**Files\FSI Clients Access Setup**" menu to publish the transformation settings.



3. Download and Install Fraser Stream Reports (client version) on a PC and register it with Fraser Stream Integration. To register Fraser Stream Reports:

   - Open "\Help\ Register" menu on Fraser Stream Reports and copy the Product ID from the registration screen.

   - Open Fraser Stream Integration on the computer where it is installed and input the Product ID and the user name in "Files\FSI Clients Access Setup" screen and click "Save" button.

4. Download and Install Fraser Stream Reports (client version) on a PC and register it with Fraser Stream Integration. To register Fraser Stream Reports:
   - Open **"\Help\ Register"** menu on Fraser Stream Reports and copy the Product ID from the registration screen.
   - Open Fraser Stream Integration on the computer where it is installed and input the Product ID and the user name in "**Files\FSI Clients Access Setup**" screen and click **"Save**" button.

5. Set in Fraser Stream Reports the same Shared Folder as the Shared Folder in Fraser Stream Integration. (See step 1 above).

6. After these steps are done Fraser Stream Reports has the same functionality as Fraser Stream Integration. If the transformation settings are created on Fraser Stream Integration then Fraser Stream Report can transform a text report in Excel, PDF, Word, XML or HTML.

3 Fraser Stream Reports (Standalone Version)

Fraser Stream Reports – Standalone Version has the same functionality as Fraser Stream Integration except batch functionality. It has to be registered on each computer where it is installed. The Shared Folder is not mandatory for this version.

4 Fraser Stream Integration Batch Functionality

Fraser Stream Integration (FSI) automates the process of transforming ASCII text reports (or delimited files) into EXCEL, PDF, CSV, HTML, XML or ADO DataSet format. FSI can automatically generate web reports using XML/HTML output or, populate either existing tables or create new database tables altogether in Access, SQL, or Oracle. Use FSI to post sales reports on the web or to populate tables for your Business Intelligence applications.

FSI can integrate Fraser Stream Reports' text file transformation capabilities with other applications and batch processes. FSI contains a simple command structure which can be included in a batch process or integrated into any Windows based application; use the command structure in your Visual Basic, C++, legacy, and ERP applications.

FSI has three distinct functions all of which can be automated using Windows Scheduler or another scheduling tool.

1. **Batch Creation**

   Create batch files that automatically select, transform, and deposit your text reports into any folder in CSV, HTML, XML, PDF, and DataSet format. Your destination folders can be individual user folders or web report folders for your entire organization.

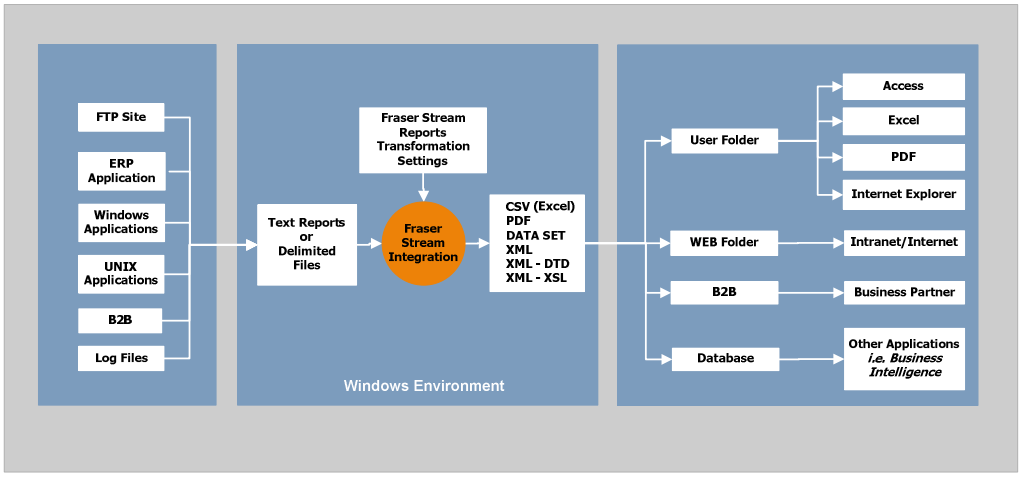
2. **Database Output**

   Text report or CSV files can be output directly into a database table. FSI provides two distinct methods of outputting data to a table. FSI can either pre-populate an existing table or generate a new table on the fly. FSI supports any database format including MS Access, SQL, and Oracle.

3. **HTML Reporting Structure**

   Create HTML page hierarchies that contain links to the folders that contain your newly transformed files. These web pages can be made available through your intranet or on the Web and secured directly through the folder security settings.

Fraser Stream Integration can automate the transformation of potentially thousands of reports daily. The following diagram outlines the various data sources that can feed Fraser Stream Integration. FSI in turn can transform and output the text or delimited file data into a variety of formats for further use throughout the organization.



FSI is able to output the original data into a folder from where it can be accessed directly, or, output it into a database table through one of two methods.

Key Benefits

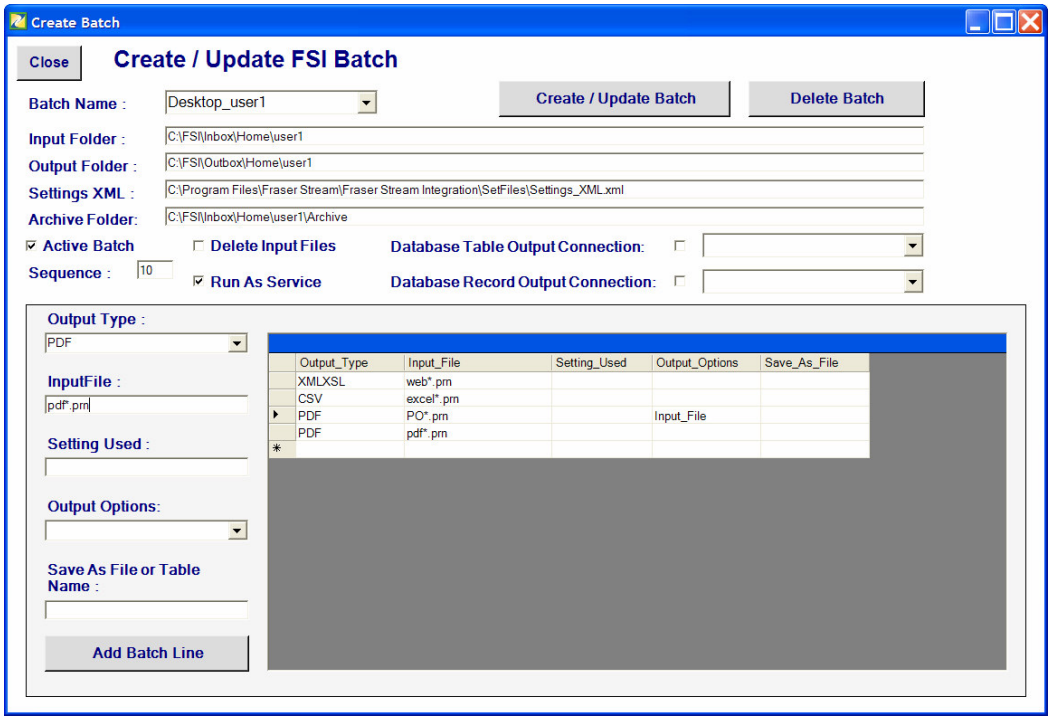
Fraser Stream Integration has all the benefits of Fraser Stream Reports plus:

1. automatically transforms the text or delimited files directly into CSV, PDF, and web formats
2. runs as a simple command that can be part of a batch process or can be integrated into other applications
3. facilitates information exchange from one application to another using XML (the standard format for application information exchange today)
4. transforms ASCII text reports into Microsoft ADO DataSet in XML format – this format can be used by ASP applications or can be exported into other databases
5. improves business to business data exchange by quickly extracting, transforming, and delivering data to business partners via CSV, XML, PDF, or Data Set format
6. creates a unified report format from different data sources
7. is quick and easy to implement – FSI does not require a great deal of technical knowledge
8. outputs data directly into database tables

a. creates a table that contains the text report's information – ideal for preparing data for use with Business Intelligence tools such as Cognos
b. stores the text report's information within a single record of a database table (as CSV, XML, or dataset) for use with other applications

5 BATCH TRANFORMATION PROCESSES

The Batch interface creates batch files that automate the text or delimited file transformation process. An FSI batch file is able to take a text or delimited file from any folder, transform it into XML, HTML, CSV or Dataset format, and deliver the output into any other folder within your network. A batch file must also be created when exporting to a database table. (Database output will be covered in the next chapter.)



The entire FSI batch structure revolves around two kinds of batch files. The **Main** batch file and the **Individual** batch files (which can be run separately or together through the Main batch file).

5.1 Individual Batch Files

The Individual batch files are created through the FSI Batch interface seen above (in the tools/generate batch on the menu bar). Each individual batch file essentially requires 3 elements:

1. A source folder
2. A transformation setting
3. A destination folder

The key is that an individual batch file can process one or thousands of text files provided they all:

- originate in the same input folder, and
- the transformed files are deposited in the same destination folder

An individual batch file can process the same text file multiple times using different transformation settings as well as transforming the file into various file formats (even with the same transformation settings) such as XML, HTML, and Dataset as long as the output is deposited into the same destination folder.

Note: If the input and output destinations change, a new batch file will need to be created.

5.2 Main Batch File

Each time a new individual batch file is generated; the command to run this file is automatically appended to a Main batch file. When the main batch file is run, every single individual batch file is also run. There are provisions to disable individual batch files from running; this will be covered in the subsequent sections.

The Main Batch file ***Main_Batch.bat*** is stored by default in:
C:\Program Files\Fraser Stream\Fraser Stream Integration\Batch

This file can be scheduled to run, or can be manually run to execute all the individual batches. Individual batches can also be scheduled to run at separate time intervals by copying the specific command from the Main Batch file and pasting into a separate .bat file altogether.

5.3 Using the FSI Batch Creation Interface – Field Explanations

Each individual batch requires an Input Folder that contains the text files, the Transformation Settings to be used, and the Output Folder where the file(s) are to be deposited.

Please refer to the previous screen shot.

5.3.1 Header Section

This section defines the input, transformation, and output paths.

1. **Batch Name**
   Enter a unique batch name in the Batch Name Combo box if you are creating a new batch process.
   Select an existing batch name is you are editing an existing batch process

2. **Input Folder**
   Provide the path to the folder that contains the text or delimited files slated for transformation. You must have access to this folder.

3. **Output Folder**
   Provide the path to the output folder where the transformed files will be deposited. You must have access to this folder

4. **Settings XML**
   Provide the path including the file name of the file that contains the transformation settings. These setting must first be defined using Fraser Stream Reports. The settings file by default will be found:
   C:\Program Files\Fraser Stream\Fraser Stream Integration\SetFiles\Settings XML.xml
   You must copy **Settings_XML.xml** file from **.\Fraser Stream Reports\SetFiles** in the above folder.

5. **Archive Folder (optional)**
   Enter a path if you want to archive each input file after transformation. FSI will automatically move the input text or delimited file into the specified folder.
   NOTE: If running FSI as a Service, either an Archive Folder must be specified or the Delete Input File box must be checked.

6. **Active Batch**
   If this box is checked, FSI will write the command to run this batch into the Main Batch file (see previous section for a definition of Individual and Main batch files)
   If the box is unchecked, FSI will still write the command to run this batch into the Main Batch file, but the command will be suspended and will not run each time the *Main_Batch.bat* file is run

7. **Delete Input File**
   If this box is checked, the input file will be deleted after each time it is transformed.

NOTE: If running FSI as a Service, either an Archive Folder must be specified or the Delete Input File box must be checked.

8. **Run as a Service**
   If this box is checked, the batch file will be included in FSI's Windows Service. When the service is activated, each batch added to the service is run automatically every 20-30 seconds. The service scans all input folders as defined in each individual batch file and processes any text files found therein. If no files are found the batch does not run. The service continuously scans the input folders for new files. This service may run indefinitely in the background while the server is running.

9. **Sequence**
   This is an integer number which defines in what order the individual batches within the Main Batch file are to be executed.

Note: The Database Record Output Connections and Database Table Output Connection fields will be covered in section 3 Database Output. These fields define the database connection string for the two methods of database output. Outputting to a database is also part of FSI's Windows Service option.

5.3.2 Detail Section

This section defines the input, transformation, and output file details i.e. names, formats, etc…

1. **Output Type**
   Use this combo box to define the type of output file you want generated by the transformation settings. Select from the following list of file types:
   a. XML – The FSI batch transforms the input file into XML
   b. XMLXSL – The FSI batch transforms the input file into an XML that has a reference to a style sheet located in the folder XSLT. The XSLT folder must be in the same output folder where the XML file is to be stored.
   c. XMLDTD – The FSI batch transforms the input file into an XML with a reference to a DTD file stored in the same output folder where the XML file is to be stored.
   d. CSV – The input file is transformed by FSI batch into a CSV file. This file can be opened with MS Excel
   e. DATASET – The ADO Dataset format is generated from the input file.
   f. PDF – The input text file is converted to PDF

2. **Input File**
   Enter the specific name of the input file or specify a pattern if you require multiple files to be transformed. If you enter **.txt* – FSI will transform all the text files in this folder

3. **Setting Used (optional)**
   If the Fraser Stream Reports transformation setting file (specified in the header section) contains more than one set of transformation settings for this particular input file(s) you must specify which settings to use. Otherwise, by leaving the field blank, FSI will use the first set of setting that match the input file(s).

4. **Output Options**
   The output options will allow you to select a naming convention for your output files.
   a. (blank) – The output file name is generated using the report name and the name of the transformation settings used.
   b. Extend – Use this option if a unique output file name is needed. The Extend option will append the (year, month, day, hour, minute, second, millisecond) to the end of the output file name as define in option *a* thereby creating a unique file name each time.
   c. Input_File – Use this option is you want the output file name to be the same as the input file name (only the file extension will differ according output type (i.e. xml instead of txt).
   d. Save_As – Use this option for a specific output file name. The file name has to be specified in Save As File parameter.

5. **Save As File**
   Enter a specific name for the output file (including file extension). The Output Option parameter must have the value **Save_As**.

5.3.3 Command Buttons

To update the individual and main batch files**:**

Add Batch Line
As mentioned in the previous section, an individual batch can process multiple text files provided they originate in the same folder and are deposited in the same place after transformation. Each batch can convert the input files into multiple file formats utilizing different transformation settings and applying different naming conventions to the final output files.

Use this button to add individual command lines to single batch. For example in the screen shot we see that the user has specified certain text files in the input folder to be processed one way, while files with the extension .prn are processed differently. A specific text file can be transformed using a particular set of transformation settings and saved with a unique name that utilizes the time the file was transformed.

Create / **Update Batch**
This button will either create or update the individual batch to include all the commands as specified in the detail section. The batch is stored in the folder:
C:\Program Files\Fraser Stream\Fraser Stream Integration\Batch\BatchDetail

This function also updates or (first time) creates the **Main_Batch.bat** batch that contains all the individual FSI batches. The Main_Batch.bat is stored in the folder:
C:\Program Files\Fraser Stream\Fraser Stream Integration\Batch

Delete Batch
This button deletes the individual FSI batch. Be sure that the correct batch has been selected from the Batch Name field.

6 Database Output

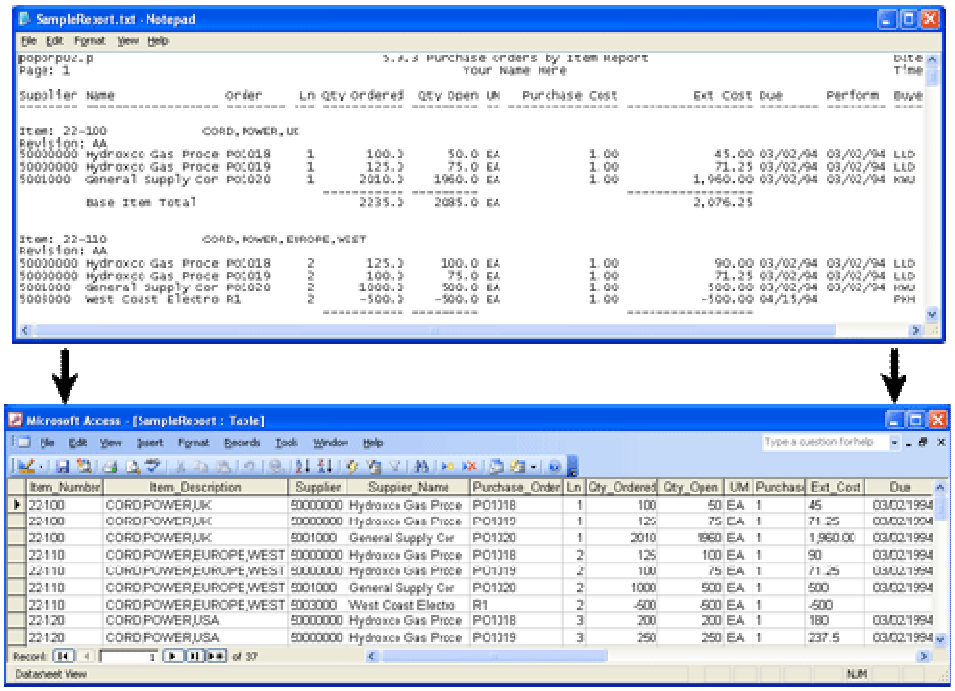
As an extension of Fraser Stream Integration's batch creation functionality, FSI can output data directly into a database table. FSI still requires that the original input text file be transformed into CSV, XML, or Dataset and deposited in an output folder. From this point FSI either appends the data to an existing table or generates a new table and populates it with the data from the output file. This process is part of the individual batch file and can be run automatically as part of FSI's Windows Service functionality.

Your data can be made available to virtually any system whether it is a custom application or your Business Intelligence solution.

Fraser Stream Integration provides two methods of exporting information to a database table.

Extract the data from a text report and load it into a database table

FSI creates a table into a database and loads the information extracted from the text file in that table. The fields names are the same as the fields generated with the transformation settings tool.
FSI provides the option to index select fields within the table to ensure that only unique records are appended to the table each time the batch is run.



Convert a text report into XML, CSV or Dataset and load it into a database record

The second method of exporting file contents to a database is far more complex and is more suitable for more advanced development. This method actually takes the entire contents of a single CSV, XML, or Dataset file and stores them in a single field within a table.

The remaining fields in each record will store information such as original text file name, file transformation date, the original contents of the text file, output file paths etc… In this scenario each report is appended to a single table record. The table structure is more suitable for staging data for further development, tracking historical changes or, for managing web content. (It is far easier to manage data within a database as compared to a folder structure)

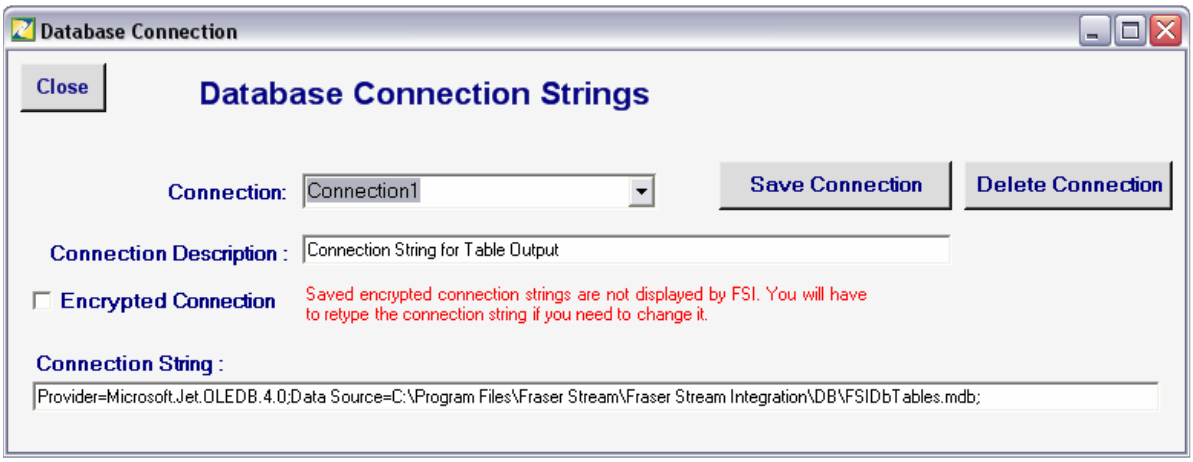
Fraser Stream Integration can export data directly into this format; however the following table structure must be strictly observed including the exact field name convention.

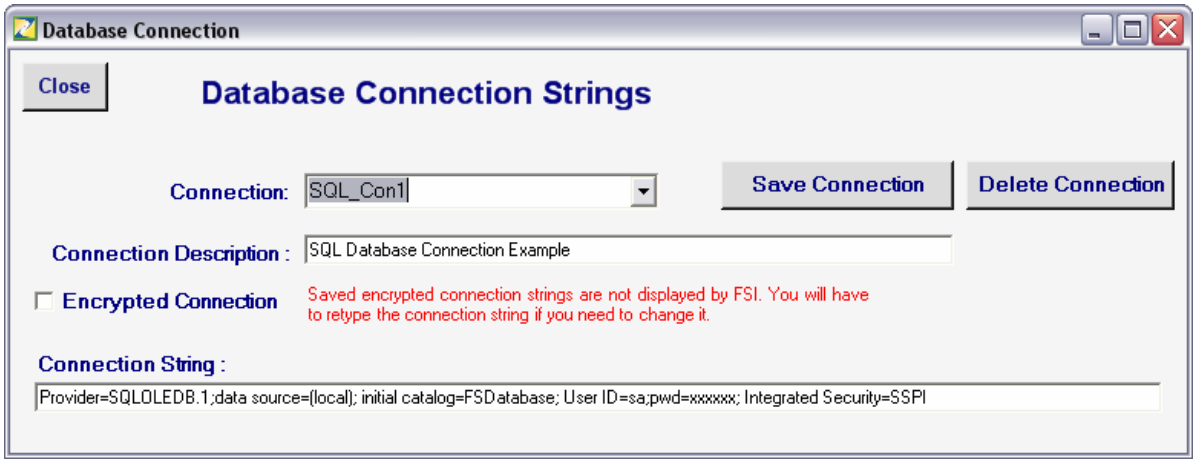
Field Name	Data Type
ID	AutoNumber
date	Date/Time
infilename	Text
infilepath	Text
infileload	Memo
outfiletype	Text
outfilename	Text
outfilepath	Text
outfileload	Memo
fsifield1	Text
fsifield2	Text
fsifield3	Text

6.1 Defining Database Connection Strings

In order to output your data to database table, you must define the connection string for the database regardless of the database output method you use.

To define the connection string, select **Database Connection** from the **Tools** menu on the main Fraser Stream Integration page.





Field Definitions:

Connection
Select a connection for your database. You can setup as many connections as necessary. By default there will be three connections along with SQL_Example connection. Connections 1 and 2 by default point to a pair of MS Access Databases included with Fraser Stream Integration and are ideal for testing purposes. (see section 3.1.1 for details)

To add more connections, simply type in a connection name i.e. *MyConnection*

Connection Description
Enter a description for you connection

Encrypted Connection
You may encrypt your connection string by selecting this check box. Your connection string will be encrypted in an xml file.

Connection String
Enter the actual connection string – see screenshot above
Use the Save Connection and Edit Connection command buttons to manage your database connections.

Once your connection has been setup the next step involves creating a batch file as described in Chapter 2 with a few minor differences.

6.1.1 Default Connection Strings

This section identifies the default database connection strings which correspond to the two different methods of outputting information to a database. Refer to section 3.2 and 3.3 for a detailed explanation of the two methods.

Connection 1: This connection string is for database output method 1 (section 3.2). By default it points to an MS Access database in the root directory under .\Fraser Stream Integration\DB\FSITable.db

Connection 2: This connection string is for database output method 2 (section 3.3). By default it points to an MS Access database in the root directory under .\Fraser Stream Integration\DB\FSIDatabase.db

6.2 Method 1: Extract the data from a text report and load it into a table

This is the standard method of importing data into a database table. Once in a database, you can connect to this information with any range of applications ranging from in house custom solutions to enterprise level Business Intelligence tools such as Cognos and Crystal Reports.

Whether exporting data to a table or to a folder, Fraser Stream Integration still requires that a batch file be created.

6.2.1 Batch Files

Refer to the previous section (Chapter 2) on the FSI Batch Creation Interface. The key difference is that you must now check the box titled:

DATABASE TABLE OUTPUT CONNECTION

Select the database connection string as defined in section 3.1. You may use Connection 1 as a default for this method of output.

Note: For method 1, you do not need to create a table in your database beforehand – as long as the database is created, FSI will create the table automatically. If you use Connection 1 as the default, the string will point to FSI's MS Access database. (found in the root directory under ***\Fraser Stream Integration\DB\FSITable.db)***

Click on the **Save Batch** button once you have finished filling in all the necessary fields.

Once the batch has been created two things will occur:

1. In the .***Fraser Stream Integration*** **Batch** folder in your root directory you will find a file called: TABLE_Batch.bat

   This is a master batch file that executes every single individual batch process that involves outputting data to a single table. Note: the process outlined in this section can be repeated multiple times for any number of text reports being outputted to table.

   With this file you can de-activate any of the individual batch commands.

2. In the .***Fraser Stream Integration*** **Batch\BatchTB** folder you will find the individual batch files that contain the commands to select, transform, deposit, then export to a table a single or a series of like text reports. For example, you may have created a batch called Sales_Order. Within this folder you will find a batch file called TABLE_Batch_Sales_Order.

   Each time this batch file is executed the following will occur:

a) Select input text file
b) Transform the file to the specified format (ie. CSV, XML, etc…)
c) Deposit the new file into an output folder
d) Delete any existing table
   * Note the option exists to continuously append new records to an existing table each time the batch is run, you may create a unique index in the output table to avoid duplicate records – see section 3.2.2
e) Create a table in the specified database (see previous comment)
f) Import the file contents into the table

6.2.2 Changing Field Types / Enabling the Append Table Option

Under Tools on the main FSI page, there exists an option called **Change Fields Types - Create Append Table**.

This page addresses two table options:

1. FSI creates a table with the information extracted from the text report. When the batch runs again the table is deleted and a new table is created. FSI will by default assign field types when the database table is created. These field types can be changed after the table has been created.

Select the connection string and the corresponding table you wish to modify from the Select Table list. A list of fields will be listed for the selected table, simply change the field properties by keying in a different field type (a list of permissible field types is listed at the top of the screen i.e datetime, varchar, integer, decimal… see screen shot on next page).

Next time when the table will be created it will have the new assigned field types.

2. Fraser Stream ETL solution (Export, Transformation and Load)
FSI creates a second table called "append table" that has the exact name as the first table but it has "*_APP*" at the end (the append table will not be deleted and recreated each time the batch is run, it will be created only if the table does not exist**). FSI provides the option to create a unique index within the append table. This index will allow you to append only those records that are unique to the index you assign.**

If the **Active Append Table** check box (see screen shot on next page) is left unchecked, FSI will create a new table each time the batch is run (deleting the old table and re-creating it along with the field type settings that have been defined in point 1).

If the **Active Append Table** is checked, FSI will create a second table within the specified database connection string. This second table will contain the same field type settings as the first table, the key difference being that the second table will not be deleted and recreated each time the batch is run. Instead, records will be appended to it or overwritten depending on the index you create (or primary key) you create. This second table will have the exact name as the first table with the addition of *_APP* at the end of the name.

Note: The first table will still be deleted and recreated each time the batch is run. It will contain a snapshot of only the latest text file. This table can serve as a validation table that will allow a comparison between your last transformation and the records added to your second Append Table.

Unique Index or Primary Key

If the Active Append Table checkbox is selected, you must create a unique field index to control how the records are appended to the table. Note the column in the screen below called *Unique_Index_Append_Table*. Simply check the field, or combination of fields you want to use for your unique index (or primary key). Each time the batch is run, only unique records will be added to your table while existing records will be overwritten.

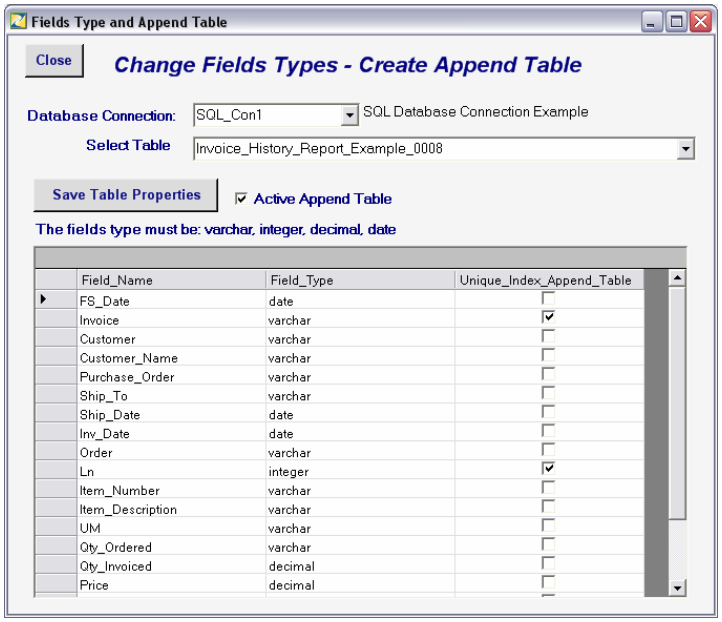
EXAMPLE: Suppose you have a sales order report that you are exporting to a table. The index you set is a combination of the Sales Order Number and the Line Number. The first time you run the batch all the contents of the text file will be added to your newly created table. The second time the batch is run FSI will check to see if the combination of sales order number and line number already exist in the table. If they exist, FSI will simply overwrite those records to ensure the most current information is in table. For example the ship quantity for a given item within sales order may have changed that day.

If the combination of sales order number and line number does not exist in the table, FSI will append the records as a new transaction to the table.

Note: After the Append Table has been created you will be unable to change Field Types for that table through FSI unless you first either change the field types manually through the database table or delete the table altogether. Otherwise the batch could fail if you change the field types after the table has already been created.

FS_Date Field

When the Append table is created for the first time, FSI will automatically create a field in the table called FS_Date. This is a date field that captures the exact time that the records are either appended or overwritten in the Append Table. This is a useful field for querying database activity and may be used to indicate when data was last refreshed.



Transaction Log

In .***Fraser Stream Integration*** *Log*\ you will find a log file called *Database_Create_Table_App.log.* This file contains a record of every Append Table transaction which includes how many records were updated (overwritten), added, or failed. An explanation is also provided for all records that fail to be processed.

Once all the field types and indexes have been set – simply click on the **Save Table Properties** command button. Then run the batch to begin the process of populating the table(s).

6.3 Method 2: Convert a text report into XML, CSV or Dataset and load it into a table record.

The second method of storing data is more complex in that the entire file output is stored in a single field within a single record in the table. This solution is more suitable for the following scenarios:

> *Data needs to be staged for various applications. This table structure will allow a developer to quickly access the XML, CSV, or Dataset contents of a file for further utilization.*
>
> *Historical data needs to be tracked for future analysis – accessing point in time information is far easier through a table than via a complex folder structure*

Web reporting is far easier to manage when the data is stored in a table. A web application need only query a record with a specific file ID – then display the XML contents stored within a specific field in that specific record.

This approach greatly reduces development time and steers clear of the traditional approach of generating a series of complex queries to return a record set.

A database connection string must be defined for both database output options. This is the first step – please refer to section 3.1.

6.3.1 Batch Files

Outputting data to a single record requires a batch file.

Refer to the previous section (Chapter 2) on the FSI Batch Creation Interface. The key difference is that you must now check the box titled:

DATABASE RECORD OUTPUT CONNECTION

Select the database connection string as defined in section 3.1. You may use Connection 2 as a default for this method of output.

Note: You need to create a table in your database beforehand - refer to section 6.3.2 for a description of the required table structure. After the initial creation, the batch will append new records each time the batch file is run.

Note: For method 2, you must create a table in your database beforehand – refer to section 3.3.2 for a description of the required table properties. After the initial creation, the batch will append new records each time the batch file is run.

Click on the **Save Batch** button once you have finished filling in all the necessary fields.

Once the batch has been created two things will occur.

Each time this batch file is executed will follow these steps:

- Transform the file to the specified format (XML, Dataset)
- Append new record to the specified database
- Loads the CSV , XML or Dataset format in the new record.

6.3.2 Table Structure to load a report converted into XML, CSV or Dataset into a record

Table Structure:

Field Name	Data Type
ID	AutoNumber
date	Date/Time
infilename	Text
infilepath	Text
infileload	Memo
outfiletype	Text
outfilename	Text
outfilepath	Text
outfileload	Memo
fsifield1	Text
fsifield2	Text
fsifield3	Text

This table may be created in any database including: MS Access, MS SQL, and Oracle.

Note: For SQL Server you may import the database and table structure from**: .*Fraser Stream Integration* \SetFiles\FSDbRecord_export.sql**

Note: You must use the appropriate field types for the specific database you are using. For example, if you are using a SQL database, you must ensure that a memo field is set to *VARCHAR* etc…

Field Description:

1. **ID**
   Record ID – this field must be the sole primary key
2. **date**
   The date and time that the text file transformation took place – when the actual batch was run
3. **infilename**
   The file name of the original text file
4. **infilepath**
   The input location of the original text file
5. **infileload**
   The entire contents of the original text file
6. **outfiletype**
   The file type of the transformed file (XML, CSV, Dataset)
7. **outfilename**
   The file name of the transformed file
8. **outfilepath**
   The output location of the transformed file
9. **outfileload**
   The entire contents of the transformed file in XML, CSV, or Dataset format
10. **fsifield1**
    Unused field
11. **fsifield2**
    Unused field
12. **fsifield3**
    Unused field

6.4 Method 3: Extract documents from a text report in XML or Dataset format and load each document into a table record.

The documents from a text report are converted into XML or Dataset format and each document is loaded into a database record. The documents stored in the database can be published onto the web in HTML or PDF format using a web interface. Fraser Stream Web interface publishes these documents onto the web with security per document type or per customer or supplier code. Using Fraser Stream document publishing functionality no programming is required to publish any documents from an ERP system on the web
Transformation setting for documents extraction:

The individual documents are extracted from the text reports and loaded into the database only if a **special transformation setting** is defined for the report.

The transformation settings contain following field names from the report header area:

1. **doc_nbr (mandatory)**
   Document Number extracted from the text file (has to unique per document like invoice, purchase order or sales order number)
2. **addr_code (mandatory)**
   Address Code - customer code, supplier code
3. **fs_field1 (mandatory)**
   Field from document header
4. **fs_field2 (optional)**
   Field from document header
5. **fs_field3 (optional)**

Field from document header

6. **fs_field4 (optional)**
   Field from document header
7. **fs_field5 (optional)**
   Field from document header

The transformation settings can contain any other fields from the text document.

6.4.1 Batch Files

To output the data to a single record a batch file has to be created.

Refer to the FSI Batch Creation Interface. The key difference is that you must now check the box titled:

DATABASE RECORD OUTPUT CONNECTION

Select the database connection string as defined in section 3.1.

Note: You need to create a table in your database beforehand - refer to section 6.4.2 for a description of the required table structure. After the initial creation, the batch will append new records each time the batch file is run.

Note: For method 2, you must create a table in your database beforehand – refer to section 6.4.2 for a description of the required table properties. When the batch is executed it will append new records for each document extracted from the text report.

Click on the **Save Batch** button once you have finished filling in all the necessary fields.

Within the .\Fraser Stream Integration \Batch folder in your root directory you will find a file called: DB_Batch.bat

Copy the DB_Batch.bat and customize it to run only the batches that are needed

Example : The batch called Batch_WebDoc.bat

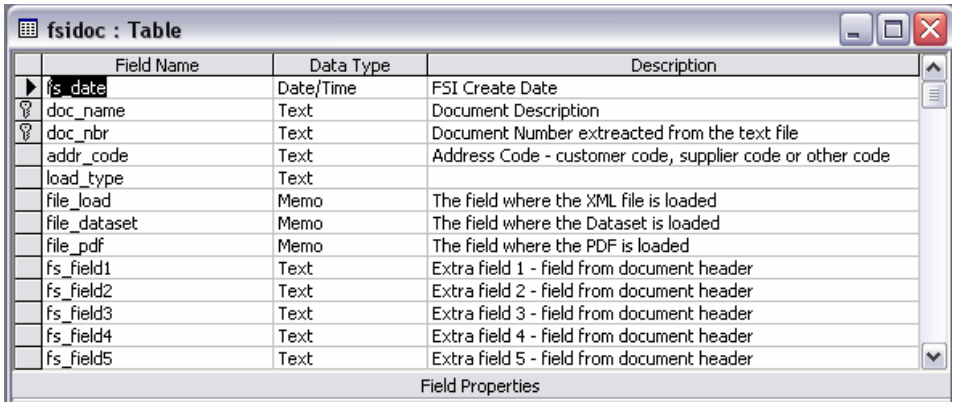
```
REM Fraser Stream Integration - Create and execute a batch
ECHO ON

CALL cd "C:\Program Files\Fraser Stream\Fraser Stream Integration"

CALL ".\bin\FS_CreateBatchDB.exe" "Batch_WebDoc" ""
"C:\FSI\Inbox\Database\DB_Web_Doc\Archive"
CALL ".\Batch\BatchDB\DB_Batch_WebDoc.bat"
```

Each time this batch file is executed will follow these steps:

- Transform the file to the specified format (ie. CSV, XML, Dataset)
- Append new record to the specified database
- Load each document from the text file into a new record in XML or Dataset format.

6.4.2 Table Structure to load each document into a table record into XML or Dataset format

Table structure:



This table may be created in any database including: MS Access, MS SQL, and Oracle.

The table needs a unique index on **doc_name** and **doc_nbr** fields

Note: You must use the appropriate field types for the specific database you are using. For example, if you are using a SQL database, you must ensure that a memo field is set to *VARCHAR* etc…

Field Description:

8. **fs_date**
   The date and time that the text file transformation took place – when the actual batch was run
9. **doc_name**
   Document Description (The "Save As" name from the Batch screen)
10. **doc_nbr**
    Document Number extracted from the text file
11. **addr_code**
    Address Code - customer code, supplier code or other code
12. **load_type**
    The conversion type (XML or Dataset)
13. **file_load** (field type can be: Text, XML, Memo)
    The field where document converted into XML is loaded
14. **file_dataset** (field type can be: Text, XML, Memo)
    The field where the document converted into Dataset is loaded
15. **file_pdf** (field type can be: Text, XML, Memo)
    The field where the document converted into PDF format is loaded
16. **fs_field1**
    Field from document header
17. **fs_field2**
    Field from document header
18. **fs_field3**
    Field from document header
19. **fs_field4**
    Field from document header
20. **fs_field5**
    Field from document header

If any other fields are need then they can be added to the table but Fraser Stream doesn't use them.

7 HTML Reporting Structure

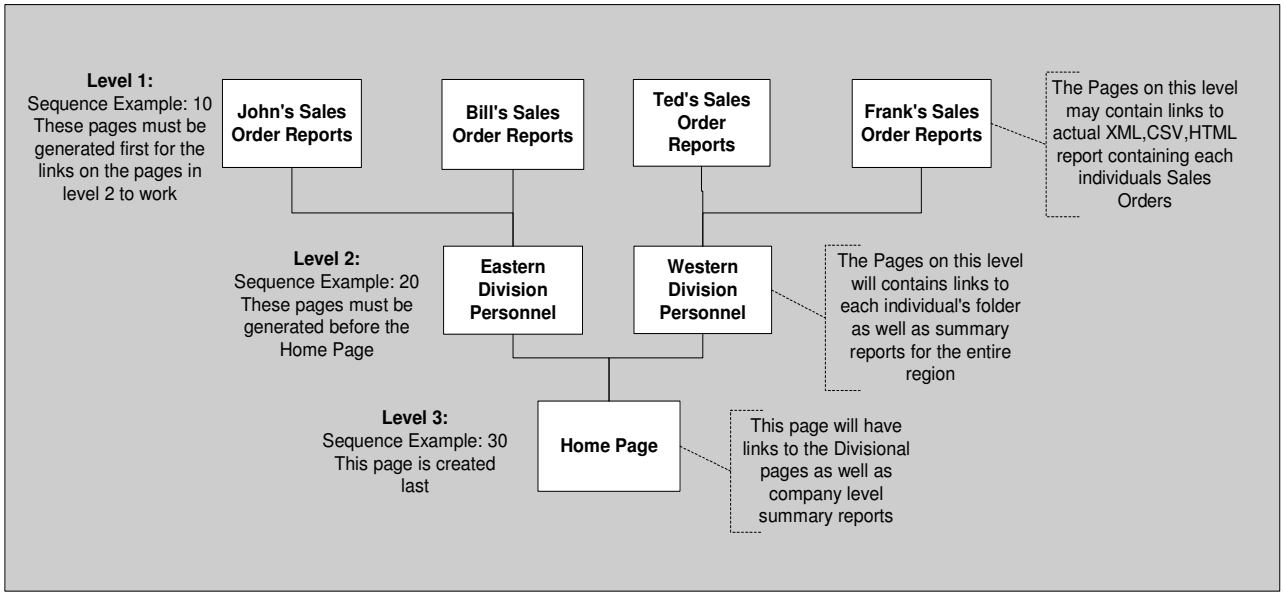
The third key component to Fraser Stream Integration is the HTML Reporting Interface. This component creates a comprehensive series of web pages that group and categorize you newly transformed files based on your folder structure.

With FSI's Batch Interface, you can transform literally thousands of files daily, depositing them into any combination of file folders on your network. With FSI's HTML Interface these file folders and the contents within them can now be made available to personnel in your company through any web browser. Security issues can be addressed at the folder level ensuring only those with appropriate access can view a folder's contents.

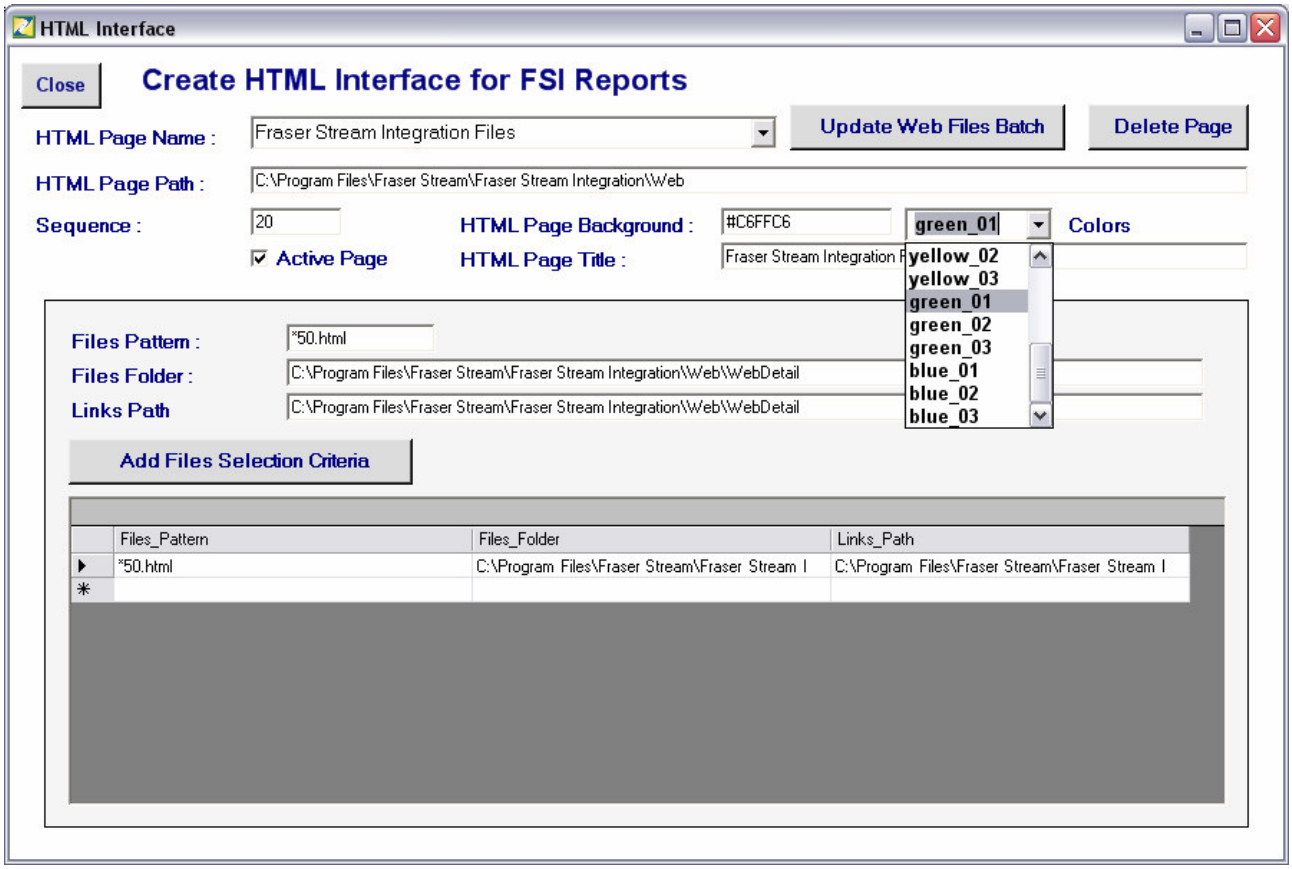
7.1 Using the FSI HTML Reporting Interface

When building the HTML page structure, you must begin at the lowest level, creating those pages that reside at the bottom of the hierarchy. The last page to be created will be the main or home page. This rule must be followed; otherwise, any links to pages lower in the page hierarchy will fail.

The following diagram is an example of simple HTML structure that can be generated by FSI.



The following interface is used to generate the HTML Reporting Interface:



7.2 FSI HTML Reporting Interface – Field Explanations

The interface is divided into two sections: Header and Detail

7.2.1 Header Section

The Header section defines the properties of the HTML page itself including the name, page path, colour, and page generation order.

1. **HTML Page Name**
   Input the name of the HTML page to be created. NOTE: Do not include the .html extension.

2. **HTML Page Path**
   Input the path of the folder in which this HTML file will reside.

3. **Sequence**
   Enter an integer value to specify the order sequence for this HTML file. Pages with a lower number will be created first.

4. **HTML Page Background**
   Select a background color from the drop-down list. It is advisable to utilize different colors to identify the various levels in the HTML hierarchy.

5. **HTML Page Title**

Enter a title to identify the types of links/files that are accessible from this page. For example, a page containing links to different production reports may be called *ABC Inc. Production Reports*

7.2.2 Detail Section

The Detail Section defines the contents of each HTML page such as links to other HTML pages further in the hierarchy, or, directly to transformed files i.e. XML, CSV, etc…

1. **Files Pattern**
   Input a pattern (i.e. *.html or *.xml) or specify an exact name of the link(s) to be included on this page.
   NOTE: The criteria will only be applied to a single folder as specified in the next field

2. **Files Folder**
   Specify the path of the folder that contains the file(s) to be displayed on this page

3. **Links Path**
   Specify the path used to create the HTML links to a list of files.

7.2.3 Command Buttons

Add Selection Criteria
Use this button to add a new line in the selection criteria window. Each HTML page can have multiple selection criteria. For example, a single HTML page can display links to all XML files in folder A, all CSV files in folder A, and all HTML pages in folder B. The newly generated page will group the links by file extension as well as by path.

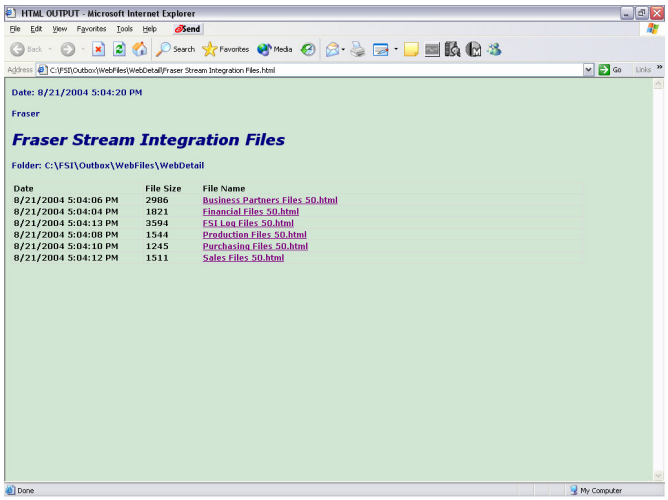
Update Web Files Batch
This command will save the HTML page with all the specified page definitions.

Delete Page
This command will delete the selected page.

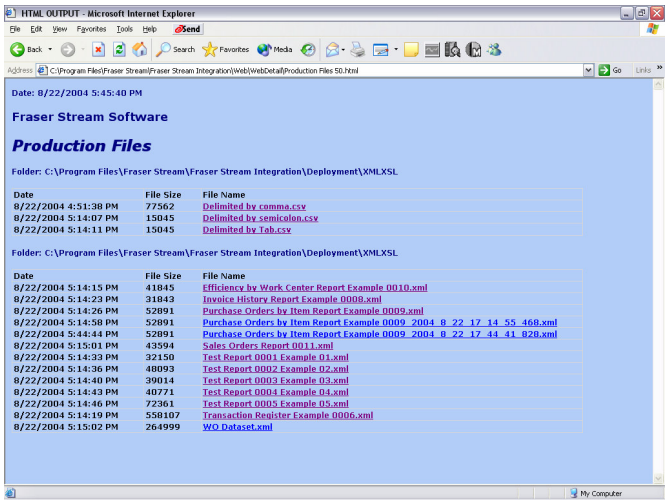
Refreshing Pages
All web pages are automatically refreshed each time the Main_Batch.bat file is run. This file not only reruns each individual batch process, it also refreshed the HTML page structure.

The following screenshots are examples of a pages created with FSI's HTML Interface. Note the two different kinds of files that are displayed (XML and CSV).

Home page:



Production Files HTML page:



8 FSI Batches and Commands

The following section covers the logic behind the batch files that drive Fraser Stream Integration. These files may be created and modified directly through the code by the developer to create customized processes if FSI's Windows Service does not suffice. It is crucial that the command structure be followed exactly to ensure that all steps in the transformation process succeed when executed.

8.1 Main Batch File

The entire FSI batch structure revolves around two kinds of batch files, the Main batch file and the Individual batch files. Each time a new individual batch file is generated the command to run it is automatically appended to a Main batch file. This file when run, will initiate every single individual batch file.

Note: The Individual Batch file essentially contains the series of commands which instructs FSI where to select the input file from, how to transform, which file output to use, and where to deposit the outputted file.

The Main Batch file Main_Batch.bat is stored by default in:

C:\Program Files\Fraser Stream\Fraser Stream Integration\Batch

The following two commands are contained in the Main_Batch.bat:

```
CALL ".\bin\FS_CreateBatch.exe" "Batch_Production"
CALL ".\Batch\BatchDetail\Batch_Production.bat"
```

The first command "FS_CreateBatch.exe" creates the individual batch "Batch_Production.bat" in the Batch\BatchDetail folder.

The second command executes this newly generated individual batch file.

8.2 Individual Batch File

The Individual Batch file essentially contains the series of commands which instructs FSI where to select the input file from, how to transform, which file output to use, and where to deposit the outputted file.

The Individual Batch file is stored by default in:

C:\Program Files\Fraser Stream\Fraser Stream Integration\Batch\BatchDetail

Example of an individual FSI batch file:

```
cd "C:\Program Files\Fraser Stream\Fraser Stream Integration"

CALL ".\bin\FS_Runtime.exe" "XMLXSL" ".\Deployment\TextReports\Work Order Status Report
0007.prn" ".\SetFiles\Settings_XML.xml" ".\Deployment\XML" "" "" ""

CALL ".\bin\FS_Runtime.exe" "XMLXSL" ".\Deployment\TextReports\Invoice History Report
0008.prn" ".\SetFiles\Settings_XML.xml" ".\Deployment\XML" "Example 0008" "" ""

CALL ".\bin\FS_Runtime.exe" "XMLXSL" ".\Deployment\TextReports\Purchase Orders Report
0009.prn" ".\SetFiles\Settings_XML.xml" ".\Deployment\XML" "" "Extend" ""

CALL ".\bin\FS_Runtime.exe" "XML" ".\Deployment\TextReports\Sales Orders Report 0011.prn"
".\SetFiles\Settings_XML.xml" ".\Deployment\XML" "" "Input_File" ""

CALL ".\bin\FS_Runtime.exe" "DATASET" ".\Deployment\TextReports\Work Order Status Report
0007.prn" ".\SetFiles\Settings_XML.xml" ".\Deployment\XML" "" "Save_As" "WO Dataset.xml"
```

The individual batch file is an executable that contains all necessary parameters for selecting, transforming and depositing the text report or delimited file. The FS_Runtime.exe command structure occurs as follows:

"FS_Runtime.exe" "Output type" "Input file" "Settings file" "Output folder" "Setting used" "Output Options" "Save As File"

8.2.1 FS_Runtime.exe Parameters Description

1. **FS_Runtime.exe**
   This executable file is the command transforms the text report. If Fraser Stream Integration has been registered on a Windows server then the command can be used in a batch or can be integrated into any other application running on that server.

2. **Output type**
   This parameter defines the type of output created by the command. The output can be CSV, XML, XML with DTD, XML with Style Sheet and ADO Data Set in XML format.

3. **Input file**
   This is an ASCII text report created by a UNIX or Windows application. Fraser Stream Integration needs read access to this file. The file can be anywhere on the server or on the network where Fraser Stream Integration has reading rights.

4. **Settings file**
   This is a file created by Fraser Stream Reports that contains the transformation rules for a list of text reports. The file name is Settings_XML.xml; Fraser Stream Integration needs read access to this file. The file can be anywhere on the server or on the network where Fraser Stream Integration has reading rights.

5. **Output folder**
   The output file created by the FS_Runtime command will be stored in this folder. The folder can be anywhere on the server or on the network where Fraser Stream Integration has writing rights.

6. **Setting used (optional parameter)**

If this parameter is not specified then Fraser Stream Integration will use the first set of transformation settings that is valid for the input file. If the transformation has to be done with a specific set of transformation settings then this parameter will identify that particular set of transformation settings to be used. This parameter is "My title" value chosen by the user when the transformation settings are generated by Fraser Stream Reports.

7. Output Options (optional parameters)

Extend
If the parameter "Extend" is specified then the file name will be extended with the date and the time when the output file is generated (the time format include the milliseconds). The extended file name is useful when the user needs to generate unique file names for the output files. Example of output file name: Test Report 0001 Example 01_2004_3_4_20_29_25_750.csv

Save_As
if the parameter "Save_As is specified then the output file name will be the same as the file name from the "Save_As_File" parameter.

Input_File
If this parameter is used then the output file name is the same as the input file but the file extension is changed according with the output file type.

8. Save_As_File (optional Parameter)
If the "Save_As" option is used in the previous parameter (Save_As under the Output Options) then this parameter specifies the output file name.

8.3 Database Output Method 1 – Extract the data from a text report and load it into a table

The following section describes the command structure for outputting data using Method 1 where a new table is generated containing the contents of the text report record by record.

The Main Batch file **TABLE_Batch.bat** is stored by default in:

C:\Program Files\Fraser Stream\Fraser Stream Integration\Batch

The table output batch is an executable that contains all necessary parameters for selecting, transforming, creating and importing the text report or delimited file into a database table. The **FS_Runtime_DB_TB.exe** command structure occurs as follows:

"FS_Runtime_DB_TB.exe" "Output type" "Input file" "Settings file" "Output folder" "Setting used" "Output Options" "Save As File" "Connection"

Each time the **TABLE_Batch.bat** is run, FSI creates a table in the specified database and imports the source data into it. If the table already exists, it will be deleted and replaced with the newly created one.

The command has the all parameters from standard individual batch file FS_Runtime.exe plus one additional parameter - "Connection".

"Connection" parameter
The "Connection" parameter is the connection code that specifies which database connection string is to be used by the command. The connection codes are created in the "Database Connection" menu.

8.4 Database Output Method 2– Convert a text report into XML, CSV or Dataset and load it into a table record

The following section describes the command structure for outputting data using Method 2 where a new record is generated in an existing table and the contents of the text report converted into XML, CSV or Dataset is loaded into a field.

The Main Batch file **DB_Batch.bat** is stored by default in:

C:\Program Files\Fraser Stream\Fraser Stream Integration\Batch

The Database output batch is an executable that contains all necessary parameters for selecting, transforming, creating and importing the text report or delimited file into a record within an existing database table. The **FS_Runtime_DB.exe** command structure occurs as follows:

"FS_Runtime_DB.exe" "Output type" "Input file" "Settings file" "Output folder" "Setting used" "Output Options" "Save As File" "Connection"

Each time the **DB_Batch.bat** is run, FSI creates a record into a table called **"filelist"** within the database specified in the connection string. The original text report or delimited file is stored in it's entirety within a single field in the record. The content of the output file whether it be XML, CSV, or Dataset is stored in a completely separate field within that record.

The command has the all parameters from standard individual batch file FS_Runtime.exe plus one additional parameter - "Connection".

"Connection" parameter
The "Connection" parameter is the connection code that specifies which database connection string is to be used by the command. The connection codes are created in the "Database Connection" menu.

8.5 Database Output Method 3– Extract documents from a text report in XML or Dataset format and load each document into a table record.

The following section describes the command structure for outputting data using Method 3 where a new record is generated into an existing table for each document extracted from a text report and loads it into that record in XML or Dataset format.

The Main Batch file **DB_Batch.bat** is stored by default in:

C:\Program Files\Fraser Stream\Fraser Stream Integration\Batch

The Database output batch is an executable that contains all necessary parameters for selecting, transforming, creating and importing the text report or delimited file into a record within an existing database table. The **FS_Runtime_DB.exe** command structure occurs as follows:

"FS_Runtime_DB.exe" "Output type" "Input file" "Settings file" "Output folder" "Setting used" "Output Options" "Save As File" "Connection"

Each time the **DB_Batch.bat** is run, FSI creates a record for each document into a table called **"fsidoc"** within the database specified in the connection string. Each document is loaded into a table record along with other fields from the document header.

"Connection" parameter
The "Connection" parameter is the connection code that specifies which database connection string is to be used by the command. The connection codes are created in the "Database Connection" menu.

8.6 Web HTML Output structure

The main **Web_Files.bat** batch file creates the HTML Reporting Interface. This batch allows you to in a matter of minutes, create a comprehensive series of web pages that group and categorize you newly transformed files based on your folder structure. The batch always runs when the Main_Batch.bat is executed.

The **Web_Batch.bat** file is stored by default in:

C:\Program Files\Fraser Stream\Fraser Stream Integration\Batch

Web_Files.bat contains the command "**FS_FilesList.exe**" that genereates the HTML pages:

```
CALL ".\bin\FS_FilesList.exe" "Production Files 50" ".\Web\WebDetail\Production Files 50.html"
```

Example of Web_Files.bat :

```
REM Fraser Stream Integration - Web Batch: Web_Files.bat
cd "C:\Program Files\Fraser Stream\Fraser Stream Integration"
CALL ".\bin\FS_FilesList.exe" "Production Files 50" ".\Web\WebDetail\Production Files 50.html"
CALL ".\bin\FS_FilesList.exe" "CSV Files 50" ".\Web\WebDetail\CSV Files 50.html"
CALL ".\bin\FS_FilesList.exe" "Datasets Files 50" ".\Web\WebDetail\Datasets Files 50.html"
CALL ".\bin\FS_FilesList.exe" "Fraser Stream Integration Files" ".\Web\Fraser Stream Integration Files.html"
```

8. PDF Output

Fraser Stream Integration converts text reports into PDF format. Initially the PDF report must be designed using Crystal Reports and the structure saved as an *.RPT* file. Once the structure has been created, Fraser Stream Integration inserts the contents of any text report into the predefined report structure, and then outputs the results into PDF format.

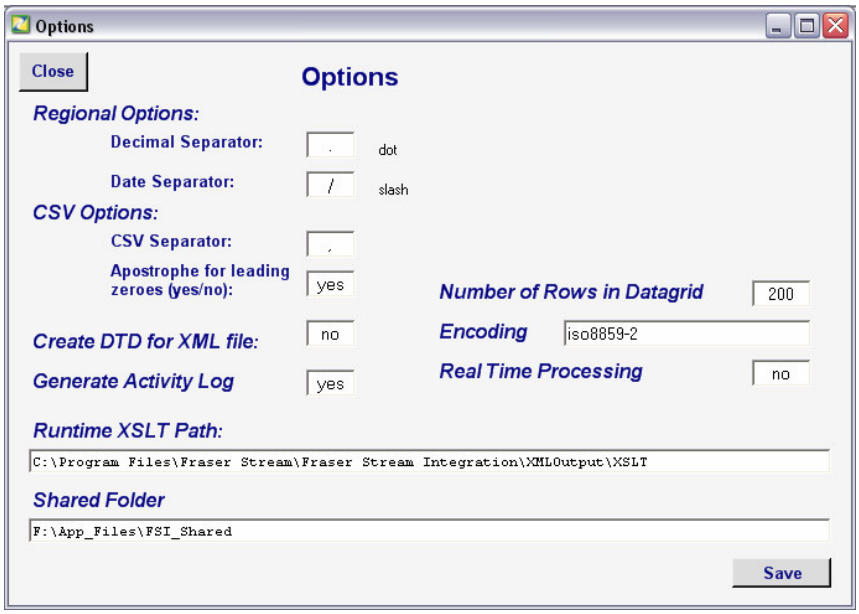
The whole process can be automated ensuring the user need only output the original text file; Fraser Stream software will do the rest, opening the PDF automatically on the user's PC.

8.7 Preconditions

The following preconditions must be met to generate PDF reports with Fraser Stream Integration:

1. Transformation Settings must be created for the text report
   (Please refer to the Fraser Stream Report Mapping Documentation)

2. Within Fraser Stream Integration, the path to the Shared Folder has to be set



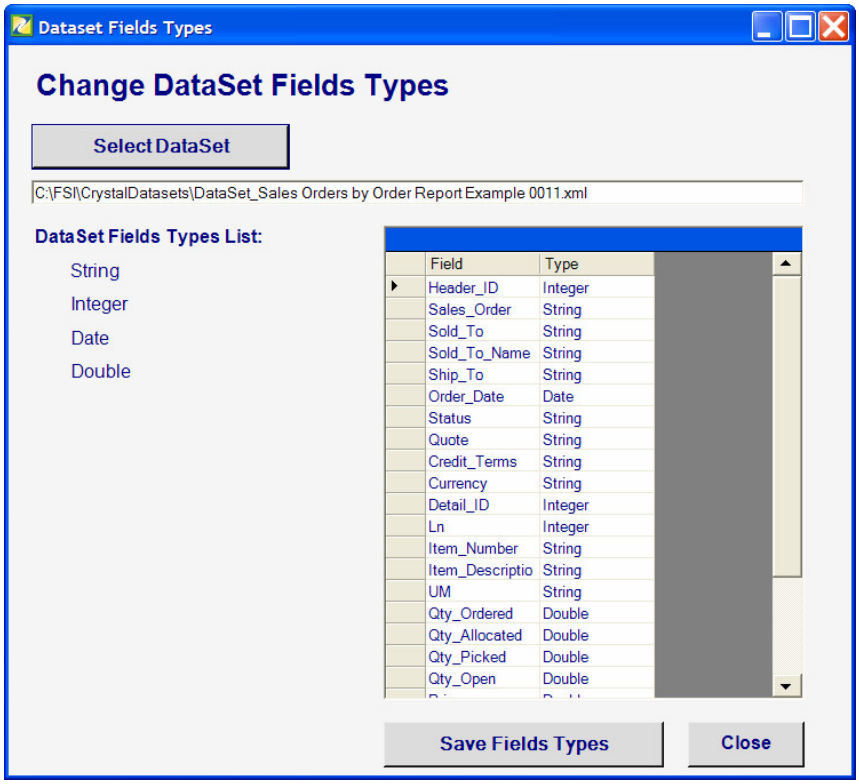
3. Click the PDF Button on the Main Screen. If the Crystal Report is not found then a message is displayed.
4. A Dataset file having the report name is created into :
   C:\Program Files\Fraser Stream\Fraser Stream Integration\Dataset
5. Verify and change the database fields types using "Change Dataset Fields Types" menu and Click the PDF Button again to re-generate the dataset

6. Copy the generated dataset into a working folder
7. Create a Crystal Report having the Dataset as data source
8. Save the Crystal Report with the same name as the Dataset name but with the extension .rpt
9. Copy the Crystal Report into [Shared Folder]\CrystalReports folder
10. Click the PDF Button to generate the PDF format

8.7.1 Change Dataset Fields Types

Fraser Stream Integration applies Dataset fields types according with the type of data extracted from the text file. For example if the Sales Order numbers in a given text report are numeric (Example: 12345) then the Dataset field type for the Sales Order field will be automatically set to numeric. If however the same text report is generated for a different division or customer where Sales Order numbers are strings (Example: SO1234) then Crystal Reports will run into trouble. To avoid data type conflicts in Crystal Reports, ensure that fields such as Sales Order are defined as strings with the Dataset file.

Using Fraser Stream Integration, select Tools\Change Dataset Fields Types from the menu to predefine the field types for a given text report.
Select a dataset from the output folder. The current Dataset field types are displayed and the field type can be changed overwriting the field type. Save the new fields types.

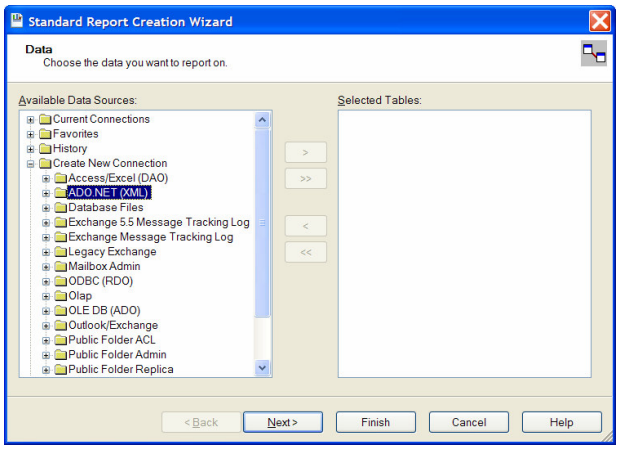


Once the dataset field types have been properly defined, the Dataset file can be opened with Crystal Reports to create the report structure for the final PDF output.

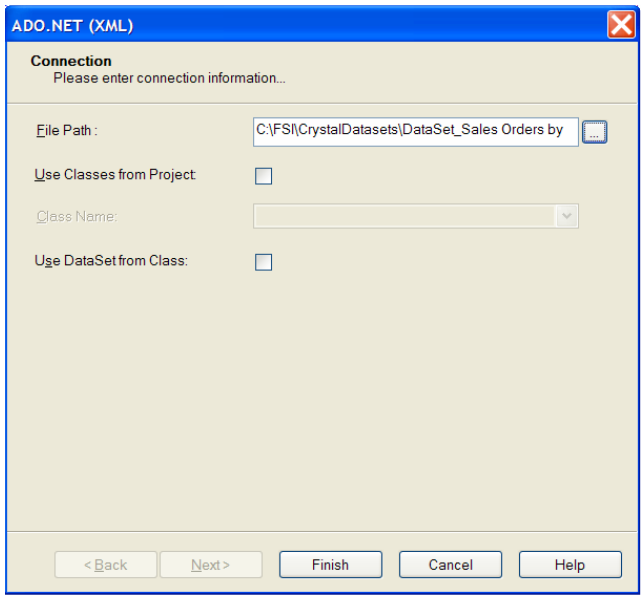
8.8 Create Crystal Reports Format

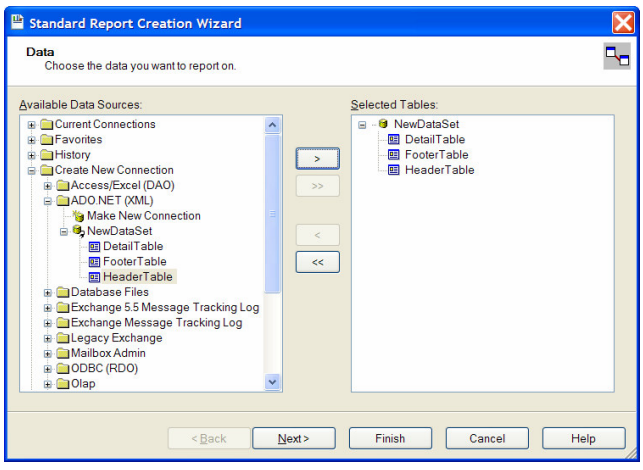
Please use Crystal Reports documentation that shows how to design the report. We cover here few basic steps about how to create the report from a Dataset source.

1. **Choose the data source type. Select ADO.NET (XML) option**

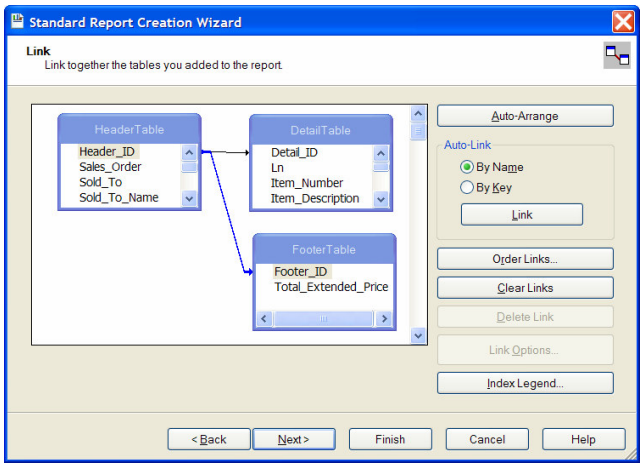


2. **Choose the data source Dataset file:**

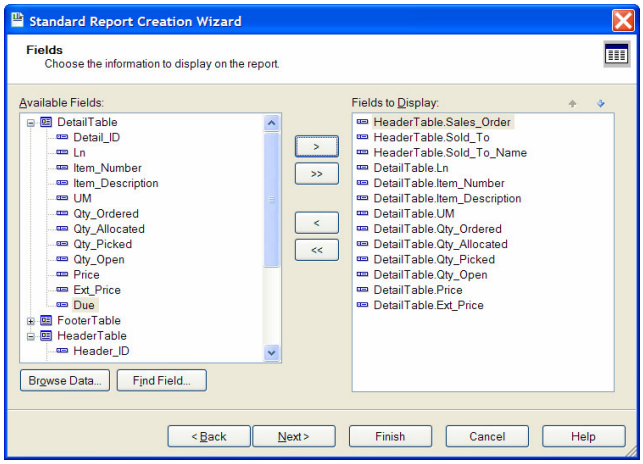




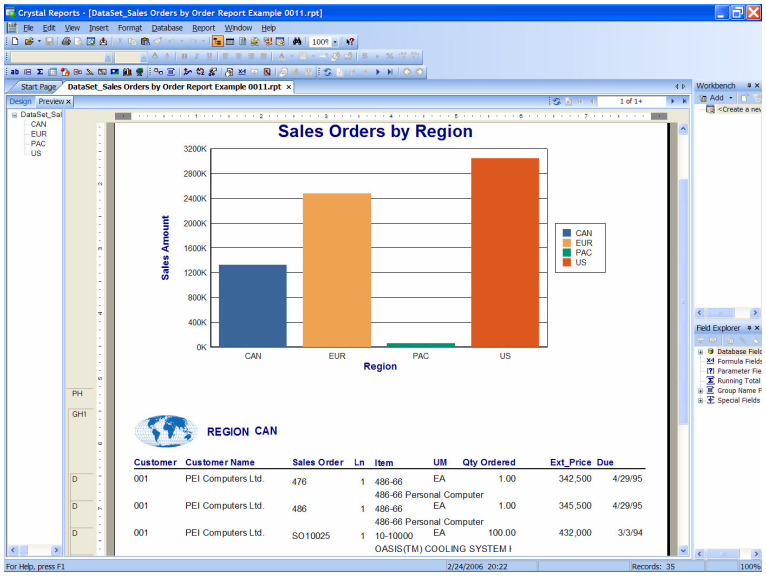
3. **Link Dataset Tables – HeaderTable, DetailTable and FooterTable**



4. **Select the fields from Dataset Tables**



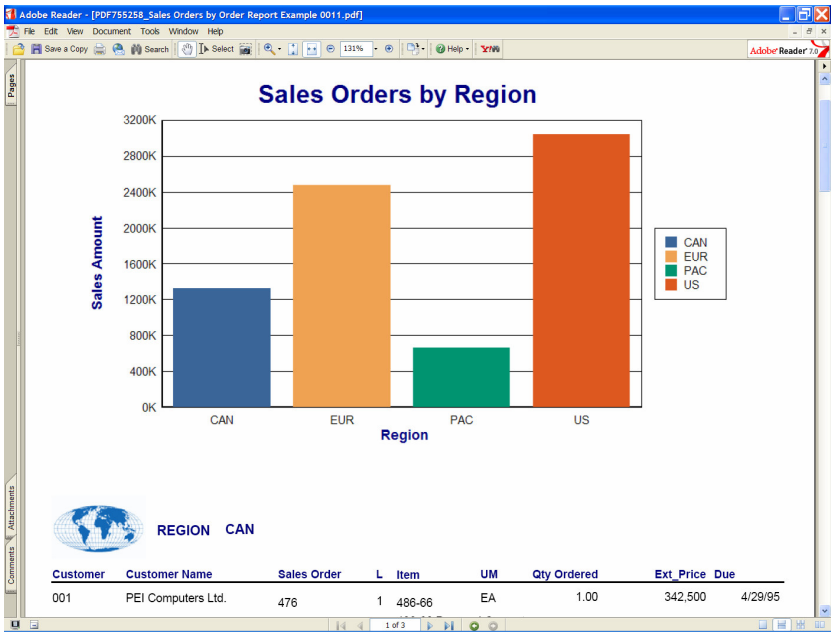
5. **Crystal Report Design Example:**



Ensure the newly created Crystal Report file name is the same as the Dataset file name but with rpt extension. Example: If the Dataset file name is DataSet_Purchase Orders Report 0009.xml then Crystal Reports report name will be DataSet_Purchase Orders Report 0009.rpt

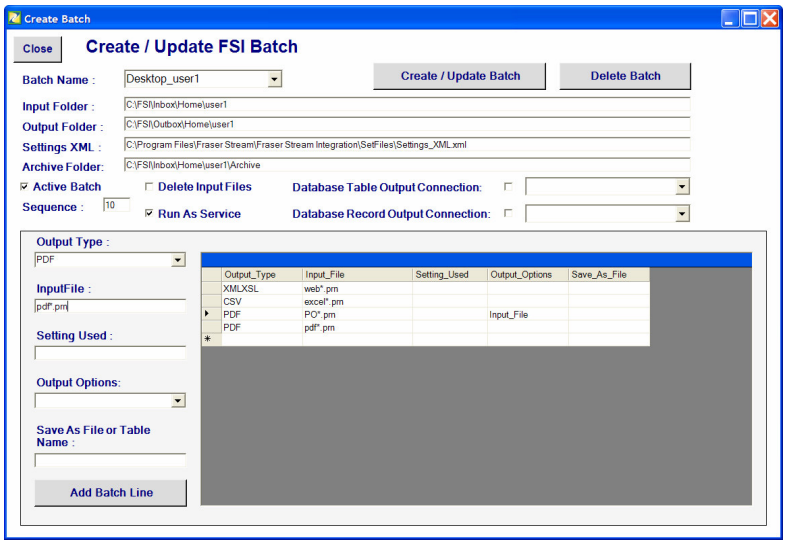
Copy the newly generated Crystal Reports file into the Crystal Reports folder in the Shared Folder

PDF Format Example:



8.9 Batch Settings for PDF Output

Once the transformation settings have been mapped Fraser Stream Integration can be configured for PDF output.



Once the original Dataset file is generated, the Dataset Field Types must be defined to ensure the report is properly compile by Crystal Reports.