



SQL Server Data Migration Approaches SSIS vs. SQL Server Stored Procedure

- Ravi Shankar Anupindi, Soumya Ranjan Das

Abstract

SQL Server integration Services (SSIS) is the de-facto standard for data migration and ETL (Extract, Transform and Load) operations in the Microsoft BI suite of Technologies. Nevertheless, SQL Stored Procedures are still the preferred choice in some of the applications for carrying out data migration activities like loading a delimited flat file into a SQL Server table.

Teams involved in data migration activities with SQL Server Suite often come across the requirement of importing data from flat files. The initial Research and Development phase lays emphasis on two of the very common approaches.

- 1) Import data using SSIS Packages.
- 2) Import data using custom SQL Stored Procedures.

Each of the above two approaches comes with its own pros and cons. Proper evaluation has to be done to select the most appropriate approach for the in-hand requirements during the analysis phase. Incorrect decisions taken initially will lead to design changes at a later stage which may incur huge rework costs and schedule deviation.

In this article, we have analyzed both the approaches for importing data from a non-delimited flat file to SQL Server tables. The parameters considered for analysis hold good for any initial research or comparative analysis as they are quite generic in nature.

Target Audience

This article will help the teams involved in data migration to zero-in on one of the above two approaches. It also throws light on some of the ways in which each of these approaches are invoked and how they might affect the decision making. This will also be helpful to teams involved in doing similar analysis for carrying out their data migration activities.

This article helps evaluating the two data migration approaches taking into consideration some of the generic data migration requirements and should not be considered as a single source of truth and similar comparative analysis should be done for other approaches in view of the requirements.

Data Migration

Data Migration is an important activity in almost every organization – arising out of constant endeavor to better the data storage and

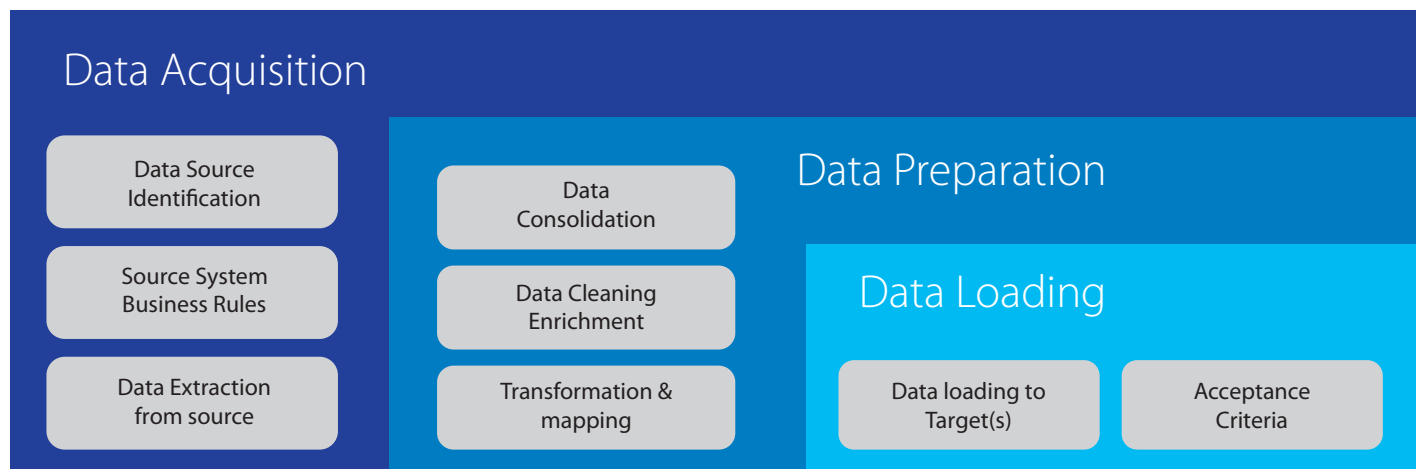
retrieval processes. The need may also arise due to the change in the technical leadership of the company or adoption of different technology stack for managing the data. Migration activity is also called for when disparate systems exchange data in different formats – say from excel files to database. The differences in the data storage mechanisms are one of the foremost reasons to initiate data migration activity.

Data migration can be of different types, some of which are listed below:-

- 1) OLTP Databases to Data warehouses or Data marts
- 2) Data Migration between two different databases – say from Oracle to SQL Server
- 3) Database Up-gradation – say SQL Server 2000 to SQL Server 2008, or from DB 2 UDB 7.0 to DB2 UDB 9.0
- 4) Excel Files to Databases or vice versa
- 5) Flat Files to Databases or vice versa
- 6) Flat Files to Excel Files or vice versa

The following strategy can be considered for most of the data migration requirements.

Fig 1 - Data Migration Architecture



Data Acquisition step involves identification of databases/systems/existing services/data extracts where source of master data or transaction data resides and finalizing rules for extracting the required subset of data.

Data Preparation step involves transforming/cleansing of the data to improve data quality.

Data Loading step involves loading the data in the target system checking for correctness.

The in-house technology stack, the source and destination data formats, together, form the most compelling factors in deciding the appropriate data migration approach. Besides, the data volume and the frequency of migration also affect the decision making. In choosing the one-time data migration approach, detailed in-depth analysis of the various approaches may not be considered and can be based on the comfort level of the team. The same doesn't hold true with migration tasks involving high frequency and high data volume. In such cases, proper in-depth analysis has to be done in choosing the most suitable approach.

SSIS vs. SQL Stored Procedure analysis for Non-delimited data file migration

In this article, we attempt to put forth the analysis, considered to migrate the data from a non-delimited flat file to SQL Server 2008 R2 database. Key challenges being migrating high volume of data on a daily basis coupled with high transformation and lookup operations within a stipulated timeframe. Two approaches were evaluated for this purpose

1. Using SSIS
2. Using custom SQL Stored Procedures.

The selection of these approaches was dependent on the following factors.

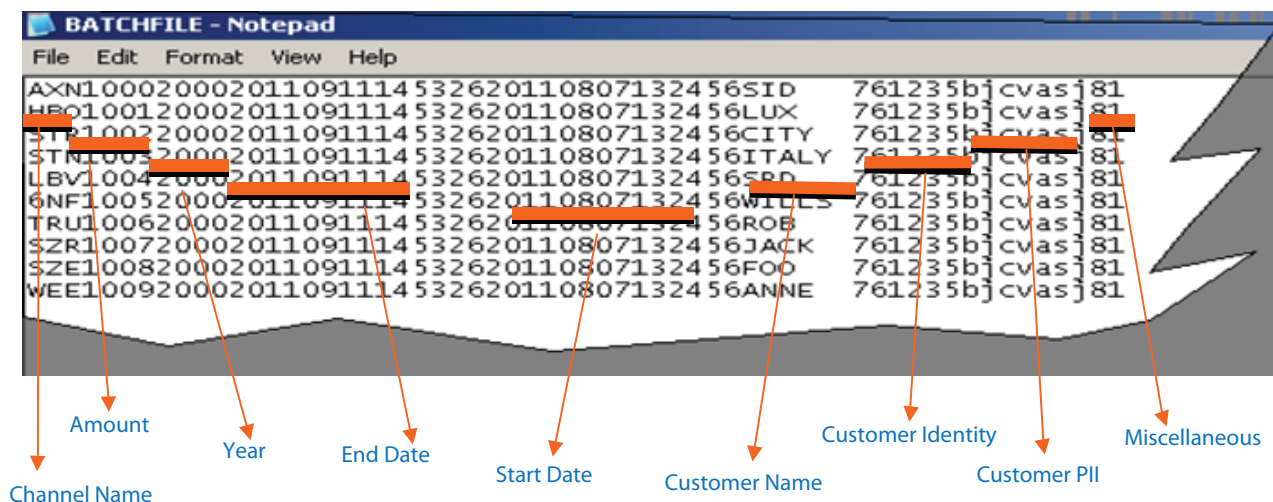
#	Attributes	Description
1	Technology Stack	<p>Our technology stack consisted of</p> <ul style="list-style-type: none"> • SQL Server 2008 R2 database • SQL Server Business Intelligence tools – namely, SQL Server Integration Services (SSIS), SQL Server Analysis Services (SSAS) and SQL Server Reporting Services (SSRS). <p>The presence of SQL Server and SSIS in the tech stack enabled us to go with a cost effective solution over other ETL tools like Informatica etc., as we need not have to procure licenses.</p> <p>Also, we could choose to write classical Transact-SQL using SQL Server.</p>
2	Destination data source	SQL Server 2008 R2 as the destination data source, helped in narrowing down the selection criteria.
3	Team experience	The team had prior experience with SSIS and SQL Stored procedures.

Before we deep dive into the comparative analysis of both the approaches, let's try to understand the source data which triggered the migration analysis.

Source Data (Non-delimited flat file)

A Non-delimited file does not have any delimiters like ";" or "" to identify individual fields in a single row of the file. Columns or fields are distinguished using the offset and length from the start point in each row of the file.

Fig 2 - Batch File containing non-delimited data



In a non-delimited file, the length and offset fields are used to identify the data fields.

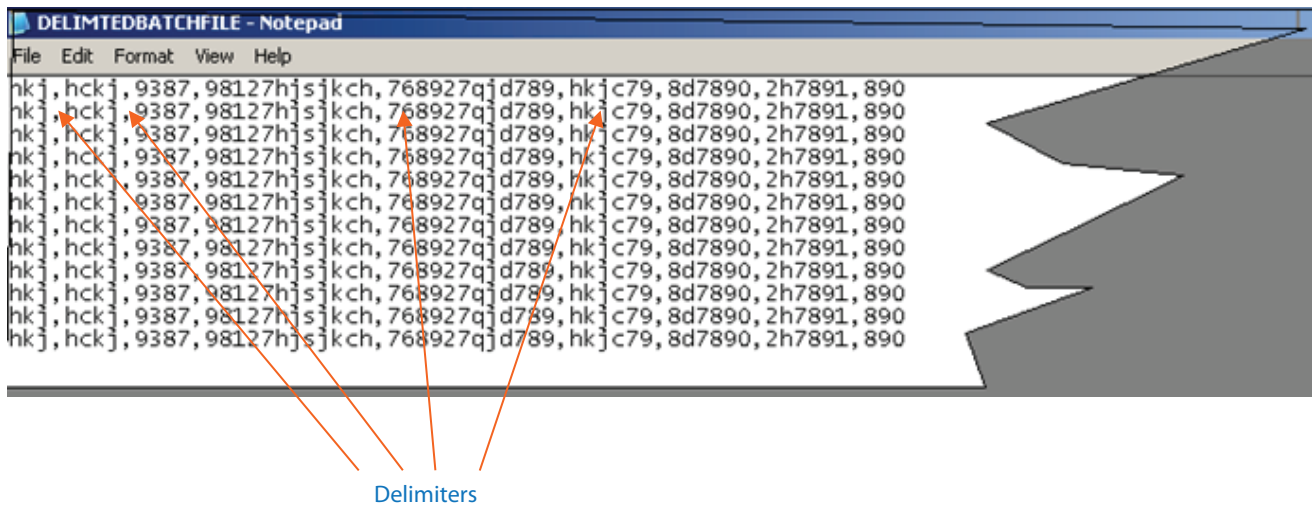
Length	Offset	Field Name
First 3 letters	0	Channel Name
Next 4 letters	3	Amount
Next 4 letters	7	Year
Next 14 letters	11	End Date
Next 14 letters	25	Start Date
Next 6 letters	39	Customer Name
Next 6 letters	45	Customer Identity
Next 6 letters	51	Customer PII
Next 3 letters	57	Miscellaneous

For example, to identify the 'Channel Name' field in the above case, we have to consider two things.

- 1) **Offset** – Starting point for identifying the field. For 'Channel Name' Offset is 0, so the starting character for 'Channel Name' field is 'A'.
- 2) **Length** – Length from the offset field that identifies the ending character of the field. For 'Channel Name' Length is 3, so the ending character is 'N'

Hence, to find the value of 'Channel Name' field we need to consider the set of characters between the offset and length fields. With offset value as 0 and length as 3 we get 'AXN' as the field value for 'Channel Name' in the first row. Similarly, other field values are computed using their corresponding length and offset values.

In contrary, a delimited file will be something like below.



Migrating data from a delimited file is much easier as compared to a non-delimited file as later involves more processing to compute the length and offset of the field values.

Data Migration Approaches

The below section provides a brief overview of how each of the selected approaches handle data migration of a non-delimited file, followed by a detailed comparison amongst them. The comparative analysis is done from following perspectives:-

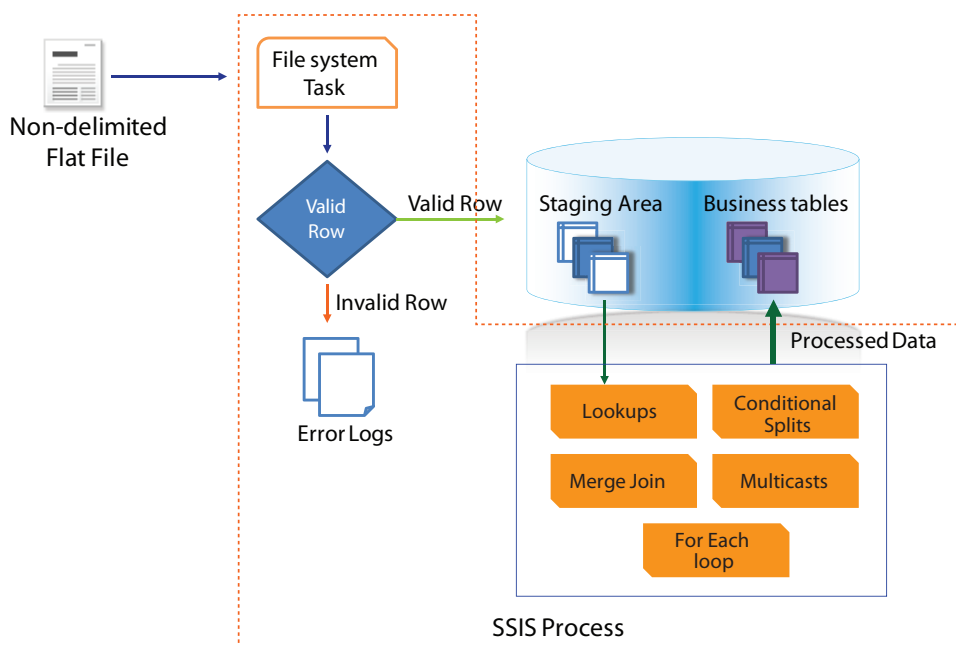
1. Generic comparison analysis.
2. Contextualized comparison analysis for Non-delimited files migration.

The SSIS Approach

SSIS (SQL Server integration Services) has become the de-facto standard for data migrations and ETL operations. It contains a plethora of components and tasks that can carry out a lot of operations ranging from backing up of databases, do merge joins, executing stored procedures, executing SSIS packages, and reading excel files and flat files to processing multidimensional cubes. The host of ready-to-use inbuilt tasks makes it a convenient tool for developing any simple or complex application in a very short time. Also, the SSIS internal engine provides out-of-the-box memory management and auditing features like error logging.

The below diagram depicts the control flow while reading a non-delimited flat file using SSIS.

Fig 3 - SSIS approach to read non-delimited flat file

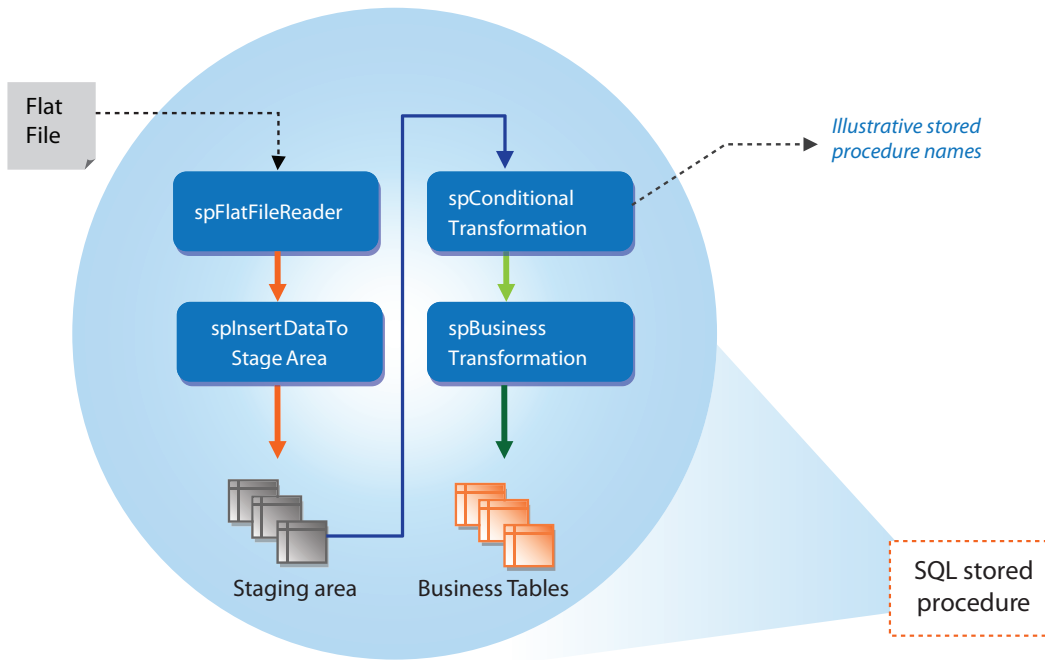


SQL Stored Procedure Approach

SQL Stored Procedures are developed using SQL Server Transact-SQL (T-SQL) programming language. T-SQL is a powerful language for developing any logic that retrieves, inserts and updates data from/into tables. Stored Procedures are a pre-compiled set of T-SQL statements that can be invoked by name. They are useful from the security point of view – in the way that specific set of users can be granted access to execute a particular stored procedure.

SQL stored procedures are being used for data migration activities in most of the applications; it is not a viable option when the data-migration involves any ETL operations. It provides high ability for customization, custom logging, and control over the logic and data management. Although powerful with a strong data type set and operators, it does not provide out-of-the-box memory management and auditing features.

Fig 4 - SQL Stored procedure approach to read non-delimited flat file






















Generic comparison analysis




The table below provides a generic comparison between both the approaches taking into consideration many of the different factors like data handling, memory management, OOTB features and others. Most of these data points come in handy to choose the appropriate data migration strategy based on the requirements in hand.



#	Attributes	SSIS	Stored Procedure	Preferred Choice
Migration Type				
1	Plain Vanilla migration	<ul style="list-style-type: none"> Not a best fit for plain Vanilla migrations which doesn't involve transformations, lookups and merge operations. Can be leveraged but is an over kill for such operations 	<ul style="list-style-type: none"> Best fit for migration activities which don't involve transformations, lookups and merge operations 	
2	ETL Support	<ul style="list-style-type: none"> Suitable for ETL nature of batch jobs 	<ul style="list-style-type: none"> Not suitable for ETL nature of batch jobs 	
Data Handling				
3	Multiple data source support	<ul style="list-style-type: none"> Provides support for heterogeneous data sources like excel files, flat files, remote databases and message queues 	<ul style="list-style-type: none"> Cannot access data from heterogeneous data sources 	
4	Non-Delimited File Support	<ul style="list-style-type: none"> Supports data feed from Non-delimited files with much ease 	<ul style="list-style-type: none"> No inbuilt operation to process non-delimited files -requires reading each record into temp table using SUBSTRING function or custom solution 	
5	Identifying and Logging Error Records	<ul style="list-style-type: none"> Data identified as error records or irrelevant records during processing can be filtered to a different data sink for later examination. Provides out of the box support for identification of records which failed to migrate or read The length and format of source data can be specified and any record that does not match, is automatically filtered out by SSIS 	<ul style="list-style-type: none"> Although error records or irrelevant records can be captured using T-SQL, it requires writing custom code. No out of the box support to identify records which may be candidates for failures. The same can be done with T-SQL using custom coding which can be error prone and need to handle lot of exceptions that may arise. Need rigorous testing to validate the same. 	

#	Attributes	SSIS	Stored Procedure	Preferred Choice
6	Bulk Load Of Data From separate servers	<ul style="list-style-type: none"> SSIS can pick up data from files or data sources present in different servers and process it 	<ul style="list-style-type: none"> It is difficult to load high volume data from sources on another server/machine using T-SQL. If the source data is on a SQL Server database, migration can be done using Linked Servers, but this process is also very cumbersome 	
Memory Management				
7	Handling Intermediate Records	<ul style="list-style-type: none"> Flat files can be used to persist the intermediate data temporarily This feature is enabled by dragging the error/redirect rows to a flat file. No custom code is needed to enable this operation 	<ul style="list-style-type: none"> Data can be stored to a SQL Server table or a flat file on the same server Custom coding is needed to support this feature 	
8	Performance	<ul style="list-style-type: none"> Comparative high performance as the need to create complex temp table, Cursor, indexing for retrieve data doesn't arise 	<ul style="list-style-type: none"> Considerable effort & time is involved in tuning the performance of T-SQL written in any stored procedure Comparative performance will be less than or equal to SSIS 	
In-Built Support For Transformations to Reduce Work				
9	In-Built support for the tasks	<ul style="list-style-type: none"> SSIS provides many OOTB tasks like File System Task, Execute SQL Task, and Bulk Insert Task etc. which reduce the coding effort in writing T-SQL for these tasks 	<ul style="list-style-type: none"> T-SQL does not have in-built support for these tasks and involves considerable effort to perform lookup operation, merge join between 2 large record sets, nested lookups, split data into more than one destination based upon one or more conditions 	
10	Parallel Execution	<ul style="list-style-type: none"> User-controlled parallel execution of data-flows is possible 	<ul style="list-style-type: none"> Considerable effort & time is involved in tuning the performance of T-SQL written in any stored procedure Comparative performance will be less than or equal to SSIS 	
11	File Handling Support	<ul style="list-style-type: none"> Provides effective file handling tasks like File System Task, FTP Task, Message Queue Task, etc. Filtering of error records is done automatically and doesn't involve custom code 	<ul style="list-style-type: none"> Supports file handling operations like BULK INSERT and OPENROWSET Involves considerable effort in filtering out error records and redirecting them to separate data sink 	
12	Modularity and Ease of work	<ul style="list-style-type: none"> It is easy to modularize and group logical tasks in SSIS When running in BIDS, it is also easy to see completion of each task one by one – be it parallel or sequential execution of tasks Graphical representation of tasks, flow and control logic provides more comprehensibility 	<ul style="list-style-type: none"> The amount of modularization which can be achieved through SQL Stored procedures is limited It is the developer's responsibility to ensure modular code which in most cases is not done Developer's can be at comfort level in using the stored procedures 	
Ease of development and Ease of Use				
13	Visual Representation	<ul style="list-style-type: none"> Provides visual representation of the "work" 	<ul style="list-style-type: none"> T-SQL is just plain code 	
14	Understanding of Flow	<ul style="list-style-type: none"> Data-flows to some extent are self-documenting. Graphical representation helps in easy understanding of the flow 	<ul style="list-style-type: none"> Need proper commenting at all places to make somebody understand what the code is doing 	
15	Ease of Development	<ul style="list-style-type: none"> Development is very easy using SSIS. Creating a data flow is just as easy as answering a few questions and the logic is written by SSIS. When a flow needs to be created, different tasks are just joined using the connector 	<ul style="list-style-type: none"> It is relatively difficult to write code for data migration using T-SQL. Structural nature of programming decreases modularity of code and increased complexity 	




#	Attributes	SSIS	Stored Procedure	Preferred Choice
Source Control				
16	Source Control	<ul style="list-style-type: none"> It is a difficult task to source / version control SSIS packages 	<ul style="list-style-type: none"> SQL Files can be versioned like Text Files in any version control tool 	
17	Changes between files	<ul style="list-style-type: none"> Even if the XML version of the packages can be controlled, it is a pain to tell what logic has changed in the SSIS package by only seeing differences between the changed XML files 	<ul style="list-style-type: none"> Differences between changed SQL files can be easily identified 	
Logging and Instrumentation				
18	Support for Logging	<ul style="list-style-type: none"> Logging and error handling is provided out of the box in SSIS 	<ul style="list-style-type: none"> Enhanced ability to throw custom errors and do custom logging 	
19	Identification of Error Records	<ul style="list-style-type: none"> Logs can be easily sent to flat files or SQL Tables based on the error output of SSIS tasks – no custom code to identify error records 	<ul style="list-style-type: none"> Probable exceptions need to be identified, and handled in Try Catch blocks in T-SQL, error messages have to be raised or logged to files or SQL Tables 	
Deployment Related				
20	Ease of deployment	<ul style="list-style-type: none"> Create Package configuration file and deploy it to any server. User can provide the server details and log in information. SSIS packages work at the application level and can interact with multiple servers at a time 	<ul style="list-style-type: none"> T-SQL stored procedures can be compiled into any servers with no overhead of configuration file Stored procedures work at only the database level and can work on only the database object of the database where it has been compiled - whereas 	
21	Invoking from business layer	<ul style="list-style-type: none"> Cannot be invoked directly from business layer like Java and has to be invoked using one of the following: <ul style="list-style-type: none"> Using XP_CMDSHELL, this runs with the Service account privileges and hence poses a security issue. Using SP_START_JOB stored procedure in MSDB database. Application User Account has to be granted EXECUTE access in the stored procedure 	<ul style="list-style-type: none"> Stored procedures can be directly invoked from business layer without any issues 	
Performance				
22	Query Execution	<ul style="list-style-type: none"> Performs better than SQL in almost all situations due to efficient memory management, tried and tested compiled query plans 	<ul style="list-style-type: none"> A SQL statement will usually outperform a SSIS data-flow when the data transform is table-to-table on the same server. In other situations, SSIS tasks are efficient 	
23	In built Caching support	<ul style="list-style-type: none"> SSIS has in-built support for caching which helps in improving performance Caching mechanism offers functionality to tweak caching from No cache mode to Partial cache mode to Full cache mode 	<ul style="list-style-type: none"> No OOTB caching support available 	
24	Tuning Activities	<ul style="list-style-type: none"> Most of the SSIS Tasks inbuilt have been tuned to perform at the best speed with optimized queries, SSIS engine's internally managed memory and disk IO. However, still some performance improvements have to be done at times 	<ul style="list-style-type: none"> Each T-SQL query written has to be scrutinized for its query plans to check if it uses the correct indexes and updated statistics and performs at reasonably good speeds 	



#	Attributes	SSIS	Stored Procedure	Preferred Choice
Others				
25	Handling high volume data	<ul style="list-style-type: none"> Easily handles high volume data by providing following OOB functionalities: <ul style="list-style-type: none"> Parallel processing in batches Easy and effective lookup operations Caching operations Bulk operations like insert and update 	<ul style="list-style-type: none"> SQL Server Database Engine does not provide any of these functionalities in-built but we can achieve these through custom code 	
26	Dependency factors	<ul style="list-style-type: none"> Requires Integration Services to be up and running – which is additional on top of the SQL Server and SQL Server Agent Services 	<ul style="list-style-type: none"> Does require the SQL Server and SQL Server Agent to be up and running 	
27	Knowledge Levels	<ul style="list-style-type: none"> Support/DEV Team requires T-SQL and SSIS knowledge 	<ul style="list-style-type: none"> Classic T-SQL Mode of development – Support/DEV requires knowledge of T-SQL only 	

Contextualized comparison analysis for Non-delimited files migration

The table below provides a contextualized comparative analysis between both the approaches taking into account very high data volume (in some millions) migration on a daily basis.



#	Attributes	SSIS	Stored Procedure	Preferred Choice
Quality of Service (QOS)				
1	Maintainability and Modularity	<ul style="list-style-type: none"> High Maintainability & less complexity as compared to the stored procedure. Provides clear separation between different tasks and modules SSIS is robust and modular to maintain - flow of data and transformation of data is easily understood and maintainable 	<ul style="list-style-type: none"> Batch job will involve multiple stored procedures (Approx.>10) which increases complexity and decreases maintainability Compared to SSIS, modularity aspect of Stored Procedures is less – in the way it is difficult to visualize the data and control flow both at run time and idle time 	
2	Security	<ul style="list-style-type: none"> To invoke SSIS packages from business layer like Java, XP_CMDSHELL utility needs to be enabled, which is by default disabled due to security reasons Though a better option than XP_CMDSHELL, SP_STAR_JOB calls for executing user to be the part of SQL Agent roles, which in general are not granted by DBA 	<ul style="list-style-type: none"> No security issues as it can be invoked directly from business layer like Java No special privileges need to enable by the DBA to invoke the stored procedure packages from the business layer 	
3	Performance	<ul style="list-style-type: none"> SSIS executes in batches of data rather than the whole data at a time. This leads to increased parallel processing and hence increased performance It handles converting into batches internally by default 	<ul style="list-style-type: none"> Calls for Custom code in SP to emulate the batching behavior of SSIS to improve performance (using CTE and loops) This emulation of the batching behavior will increase code lines, complexity and testing effort 	

#	Attributes	SSIS	Stored Procedure	Preferred Choice
OOTB Features				
4	File Operations, High lookups and handling Large Volume of data	<ul style="list-style-type: none"> Non delimited input text files can be easily imported using file connection in SSIS Multiple lookup operations are involved between the file processing and final insert or update operations. It is easy and effective to perform lookup operation through SSIS Can easily handle high data volume for lookups, which will increase in the future Due to the expected large volume of data, SSIS is a good option as there are reusable components to do the loading and transformations 	<ul style="list-style-type: none"> Need considerable coding effort to read the non delimited text file and import into SQL tables. It requires reading each record into temp table and use SUBSTRING function on each record to separate the data based on length Stored procedure is not the right approach for handling the high volume of lookups as it can lead to performance issues (Lookups will involve joins on multiple string columns) Simulate lookups using loops in stored procedure which can result into more processing time. It involves more development effort as well as testing effort Can become the bottleneck when the data volume for lookups increases in future 	
27	Bulk operations, Caching and Error Handling	<ul style="list-style-type: none"> SSIS offers caching mechanism which can be changed to "No Caching" at a later stage to save primary memory space as required Instrumentation details in SSIS (Auditing/ Error logging) are very detailed and OOB, as compared to SP wherein you have to write custom code Operations like bulk update and insert can be easily done using SSIS 	<ul style="list-style-type: none"> No OOTB caching mechanism support in stored procedures which can be leveraged 	

From the above qualitative analysis perspective, SSIS package seems to be the more compelling option. Before nailing down the final approach we performed a quantitative check of both the approaches with high data volumes, the results of which are shown in the table below.

Data Volume (in Millions)	Time taken (in hours)	
	SSIS	SQL Stored Procedure
1	0.5	2
5	1	4
10	2	6
20	3	8

Following are our test server configurations

#	Server Specifications	Details
1	Operating System	Windows Server 2008 Enterprise Edition SP2 64-bit
2	Processor	Intel Quad Core Processors 3.00 GHz
3	Primary Memory	4.00 GB
4	Database Server	SQL Server 2008 SP2 Enterprise Edition (64-bit)

In the present context considering both qualitative and quantitative aspects, we found SSIS as the more suitable approach compared to SQL Stored Procedures. We recommend the teams to follow similar analysis pattern and choose the most appropriate data migration approach according to their requirements.

Conclusion

This article showcased the comparative analysis between two of the many data migration approaches viz. SSIS and SQL Stored procedure to migrate data from a non-delimited flat file to SQL Server tables. It provided a brief understanding of the non-delimited files, SSIS and SQL stored procedure approaches followed by detailed generic and contextual comparative analysis between them. Dry run was conducted on both the approaches with high data volume and completion time was considered along with the correctness of migrated data.

We hope this article will help the teams to quick start with the decision analysis and choose the appropriate data migration approach.

About the Authors

Ravi Shankar Anupindi is a Technical Architect with Manufacturing unit, Infosys. He has around 11 years of IT experience. He has been the active member of many CoEs and his main areas of interest include performance engineering and Cloud computing. He has been involved in data migration activities in the recent past. He actively participate in Infosys internal forums as well as external forums like IBM Developer works for knowledge gaining & sharing.

Soumya Ranjan Das is a Technology Lead with Manufacturing unit, Infosys. He has around 5 years of IT experience. He has worked extensively in SQL Server 2005/2008 and has been involved in data warehousing and BI projects in the past – mainly with the Microsoft BI Suite of Technologies. His interests are in SQL Server, SQL Server Performance Tuning, Dimensional Analysis, SQL Server Analysis Services, Cubes and MDX queries.

REFERENCES

1. SQL Server 2008 Books Online
2. SQL Server Integration Services Tutorials
3. SSIS Icon, SQL Stored Procedure Icons and Other Icons
4. SQL Server Learning Center
5. Other Websites – Microsoft Technet, StackOverflow

ACKNOWLEDGEMENT

The author would like to acknowledge the contribution of Saumitra Bhatnagar (Project Manager, MFGADM) and Venkataramanan N. Baskaran (Senior Technical Architect, MFGADM) for their support and guidance to the article and extending critical inputs to achieve the current structure.

About Infosys

Many of the world's most successful organizations rely on Infosys to deliver measurable business value. Infosys provides business consulting, technology, engineering and outsourcing services to help clients in over 30 countries build tomorrow's enterprise.



For more information, contact askus@infosys.com

www.infosys.com

© 2012 Infosys Limited, Bangalore, India. Infosys believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of the trademarks and product names of other companies mentioned in this document.