THE DATA WAREHOUSE BACKROOM
- AN INTRODUCTION TO THE MICROSOFT 2008 SQL SERVER INTEGRATION SERVICES

Is5/IV2014 DATA WAREHOUSING
VERSION 1 AUTUMN 2010
BY JOAKIM SNYGG

Department of Computer and Systems Sciences (DSV)
Stockholm University (SU)
INTRODUCTION

This document will introduce you to how transactional data is extracted, transformed and loaded into a data warehouse. You will get an overview of the extract, transform and load process, exemplified in the Microsoft BI/DW platform. Additionally, you will have the opportunity to get some hands on data warehousing experience through a step by step tutorial.

We will discuss the Integration Services in Microsoft SQL Server. In Figure 1 it is shown how this service relates to other Microsoft SQL Server services and to the data warehouse architecture.

![Diagram of the Microsoft BI Platform in relation to the DW architecture.](image)

PURPOSE AND DELIMITATIONS

The purpose of this document is to provide students with the theoretical background and introduction to the Microsoft BI/DW tools that are needed to getting started with the IS5/IV2014 course assignment. Central terminology and concepts of relational databases is a prerequisite and will not be explained in this document. This is also the case for central terminology and concepts of multidimensional modeling.

OVERVIEW

This document is structured in four chapters. You can read the chapters in any order you like.
The first two chapters present the Microsoft SQL Server Integration Services (SSIS), which is a part of the Microsoft BI/DW platform:

- *The Microsoft toolset* which presents the tools you are going to use in the assignment.
- *The SQL Server Integration Services (SSIS) Environment:* This chapter will give you an overview of the graphical interface you will use working with SSIS. This chapter also provides a general outlook over the SSIS architecture and its core components.

The next chapter introduces the concept of the Extract, Transform and Load (ETL) process and how this process is supported in the Microsoft environment.

- *Data Staging and the ETL Process:* This chapter will provide an overview to the ETL process in general with common techniques, and functionality, that are being used within the industry. The theoretical background is exemplified using the SSIS Environment. This chapter will help you to better understand the purpose, performance and behavior of the Integration Services when you do the tool exercises in the last chapter of this document.

The last chapter provides step by step tutorials in the SSIS and the SSAS environments:

- *Tool Exercises:* This chapter will familiarize you with the data staging area and cube analysis.

If you want to start with the theoretical background and then do the exercises or vice versa or read them more simultaneously, is entirely up to your preferences.
When working with ETL and OLAP in the Microsoft DW/BI platform, two tools are central: The Business Intelligence development Studio (BIDS) which is used to design your DW/BI solutions and the SQL Server Management Studio (SSMS), which are used to manage the data and the databases.

**SQL SERVER BUSINESS INTELLIGENCE DEVELOPMENT STUDIO (BIDS)**

The SQL Server Business Intelligence Development Studio (BIDS) is a development tool in Visual Studio environment where you can design your ETL solution. You will find BIDS at Programs → Microsoft SQL Server 2008 → SQL Server Business Intelligence Development Studio in the Windows start menu.

![Figure 2: The BIDS environment](image)

The main parts of the BIDS interface are shown in Figure 2.

- **Designer Area**: In the center of the screen is the yellow designer pane. It is here you drag and drop items from the toolbox pane.
- **Package views**: At the top of the windows are tabs that can be used to change package views. In figure 2 the control flow view is active.
- **Toolbox**: The toolbox pane is shown on the left side of the working area. It contains all objects and components you might need in designing your ETL solution. The items shown in the toolbox are dependent on which view that is shown in the designer area. In the figure, the items in the toolbox pane are the control flow items. (These and other items will be discussed later in this document)
Solution Explorer: On the right hand side is the Solution Explorer pane. This pane shows the files that are included in your project.

Properties: The property pane in the lower left of the screen shows the properties of a highlighted item.

Connection Manager Area: Below the designer area is the Connection Manager pane which is used to manage external sources like connection to databases.

SQL SERVER MANAGEMENT STUDIO (SSMS)

To manage databases you will use the SQL Server Management Studio (SSMS). In the context of SSIS it is here where you will manage and monitor your running packages. You will find BIDS at Programs → Microsoft SQL Server 2008 → SQL Server Management Studio in the Windows start menu. When you open SSMS you must first connect to a database engine.

Object Explorer: In the left side of the window is the Object Explorer pane where you can navigate through the databases on the server. By right clicking on objects in the Object Explorer you can manipulate the databases.

New Query: To execute a SQL statement in SSMS, you click on the icon New Query, and write your query in the query pane.

Execute: You run the command by clicking on the Execute button (marked with a red exclamation mark).

Results & Messages: Results of SQL queries are shown in the results pane, the tab Messages shows the log

Status list: Shows data like nr of rows affected, time to execute and name of database and database engine.

Figure 3: Executing SQL statements in SSMS
You can access the SQL Server 2008 Books Online via the help menu if you want to learn more about SQL server services and the tools. There is a search function in the help menu where you can access more information the functionality in SQL Server 2008 and the T-SQL syntax for querying the database.

**THE COMPONENTS OF SSIS**

When working in BIDS and management studio you will use solutions, projects, packages, data sources and views.

![Figure 4: Create new project in BIDS](image)

- **Solution**: When designing a data warehouse in BIDS you will create a solution. A solution is the highest level of containers in BIDS. It can contain both ETL projects from SSIS and from other SQL Server services such as Analysis Services projects.

- **Project**: A project holds work objects with a common usage. A project uses data sources as a connection reference to the data that is stored outside the package, and thereby enables the package to access data stores. A package can also contain data source views, which only contain selected objects from a database. A data source view can also hold aggregations and other calculations, new relationships between tables, and queries or apply filters to the data.

- **Packages**: Apart from Data sources and data source views projects contains packages. A package stores all objects and metadata that are needed to process a data migration job. SSIS Packages are used to merge data from heterogeneous sources into SQL Server, to populate data warehouses and to automate tasks.
Containers and Objects: There can be a wide variety of objects and components in a package. To extract data from different sources source-adapter objects are used. To transform data, task objects are used and to load the data, adapters and destination-adapter objects are used. Furthermore, precedence constraints can be applied to order the objects in branches and sequences. The objects can also be grouped into containers where iterative processing, such as looping through data sets, can be applied.

Variables: There are two kinds of variables in SSIS: User-defined and system variables, the variables can be accessed both by SSIS objects like containers, tasks and event handlers, but can also be used by script tasks and components.

Process Control Components: The package process is divided into several process control components:

- **The Control Flow Component** is the highest level in the package process. It is used to organize and arrange the activities in an ETL process. The containers in the control flow often have precedence constraints attached to them to specify when and under which circumstances the individual tasks should be performed.

- **The Data Flow Component** is where you manage the data, by retrieving, merging, splitting and/or altering data from different sources. The data flow component handles a stream of data transformations.

- **The Event Handler Component** deals with special events during the process, such as managing errors and/or anomalies.

## THE SSIS PIPELINE ARCHITECTURE

When you develop SSIS packages you work against two engines: The SSIS Runtime Engine and the SSIS Data Pipeline Engine.

*The SSIS Runtime Engine* manages packages and the execution of infrastructure; it does not move or manipulate the data itself. The runtime engine manages Control Flow components such as sending out connection requests and connections to servers managing ftp, SQL et cetera. It sends out notifications and coordinates the data flow task.

*The SSIS Data Pipeline Engine* moves and manipulates data through a series of separate streaming data transformations. Each data flow task begins with one or more source component that extracts data via connection managers. Then the extracted data is passed between a series of transformation
components and finally destination components places the transformed data into data stores. (See figure 5).

![Diagram of data flow](image)

**Figure 6: Example of a data flow.**

Executing the data flow on a separate engine enables a buffer-oriented pipeline. The SSIS pipeline architecture stream all data transformation tasks, from source to destination, into a single flow, which is kept in the memory buffers. This **SSIS Data pipeline architecture** enables the data throughput of the process to be enhanced. Some of the factors that achieve this are the following:

- Slow operations against physical memory are avoided by only keeping data in the memory buffers.
- Execution threads can work independently and in parallel as generally each data source has an individual thread of execution, that uses separate memory buffers.
- Redundant data copying is avoided by reusing the memory buffers throughout a data flow.
**DATA STAGING AND THE ETL PROCESS**

In the Data Warehouse data is extracted, transformed and loaded in the *Data Staging Area* (DSA) of the Data Warehouse backroom, in order to be further processed into a format more accessible for Business Intelligence purposes. Besides migrating data many other tasks are performed in the DSA: Data quality is improved, by correcting missing or duplicate data, and removing errors and faults, data confidentiality is measured and documented and metadata is collected stored. The ETL process also captures the flow of transactions so it can be reviewed and be used in troubleshooting later, it adapts data to be used and compared with data from many sources.

**ETL PREPARATIONS**

Before you can start to extract data you need to decide what data should be extracted and how the data should be processed. This is done by: 1) set up a high level map; 2) commence data profiling; 3) design attribute mapping models; and 4) develop a physical ETL design.

**HIGH LEVEL MAP**

The first step in an ETL project is to set up a high level map, documenting the whole ETL strategy. Start by focusing on the business requirements and understanding of the source business systems. A sandbox source system should be set up to support planning and the ETL preparation activities.

In the SSIS environment the **SSMS tool** is used for this.

**DATA PROFILING**

To be able to set up an ETL process you must be assured that you have data that meet the requirements for data quality and business rules. This process is known as data profiling. Data profiling tools are used to analyze the data sources for missing, corrupt, and duplicate values, as well as to ensure adherence of the data to constraints and business rules.

During data profiling it is decided whether to proceed with the ETL process or whether it is better to cancel it, due to lack of good quality or quantity data. Data profiling identifies quality issues whether they can be dealt with during the ETL process, or whether they have to be solved prior to the ETL process. Data profiling also provides information on unanticipated business rules, hierarchical structures and foreign key relationships.

![Data Profiling Task](image)

*Figure 7 The data Profiling task in BIDS*

In the SSIS environment a **data profiling tool** is used to guide and support the data profiling process.
ATTRIBUTE MAPPING

The next step is to describe a source to target mapping that describes how the data in the source attributes are being mapped to the target attributes and in what order they should be transferred. Data also might need to be staged during the process, and this temporary data storage also needs to be mapped in a similar manner. Attribute mapping usually consist of the steps presented below.

1. List the attributes for a dimensional table.
2. List the relevant attributes in the Source1 database.
3. Map the Source1 attributes the attributes in the dimensional table.
5. List the relevant attributes in the Source2 database.
6. Map the Source2 attributes the attributes in the dimensional table.
(Etc... if more than two sources are extracted)

Figure 8: Example of an Attribute Mapping Model.

Figure 9: Mappings in the OLE DB Destination Editor.
In the SSIS environment the attribute mapping model is defined inside different adapters, for example the **OLE DB Destination Editor**, as seen in Figure 9.1.

---

**UPDATE STRATEGY**

You will need to determine a strategy for updating the data warehouse. Important issues are how often you will update or add new data into the data warehouse. Is it being updated on a daily basis or is it sufficient with updates once a month.

The issue of historical data also needs to be considered. The amount of data can grow and be hard to handle if historical data is to be loaded for each table, but historical data is often an essential part of the analysis. There are many techniques to manage change in a data warehouse. Rapidly changing dimensions can be managed by creating a new table where the rapidly changing attributes can be stored. When it comes to slowly changing dimensions (SCD’s) three techniques to be used separately or in hybrid combinations, are proposed by Kimball:

- Type 1: Overwrite value and do not keep historical data
- Type 2: Add a dimension row that will store the changed value
- Type 3: Add one or more dimension column to keep old values

[Kimball et al., 2002]

In the SSIS environment the **Slowly Changing Dimension Wizard** is used to implement SCD techniques. Kimball’s SCD strategy type 1 is called Changing attribute, type 2 is called Historical attribute. Each column in a dimension can be assigned either SCD type 1, SCD type 2 or you can choose to regard all changes as errors, i.e. if a record has a changed attribute this will be handled as an error and be passed through the error output from the **Slowly Changing Dimension** task.

Type 3 changes are not directly supported. It requires manual changes of to the dimensional table and the data flow. In order to apply Type 3 strategy first one or more columns have to be created in order to store changed values. Note that the attribute mapping model needs to be adjusted, to reflect the proper transfer of data values between attributes. This process is not yet automated and requires definitions of SSIS expressions in the **Derived Column transformation task**.

---

**PHYSICAL DESIGN**

The physical design of the ETL process is essential to the performance and thereby the usability of the data warehouse. Other decisions that needs to be taken is data partitioning (i.e. how the data will be divided on the physical disks), which extract technology that will be used and how distribute dimensions across multiple servers, and the use of data marts. A common architecture can be seen in figure 10. In this solution all data is first extracted and processed in a working area, which contains staging tables. When ready, the data is exported to a common data warehouse. In addition a number of data marts are created and populated with data from the common warehouse. Note

---

1 Different adapters are defined to support different types of destinations, e.g. OLE DB Destination adaptor for supporting export of data to databases, Row File Destination adaptor for supporting data export to temporary staging tables. This is further described later in this document.
that other architectures also are possible. For instance, staging tables and data marts are not always implemented.

![Diagram of data warehouse architecture with data marts on different locations]

**Figure 10: Data warehouse architecture with data marts on different locations**

Physical design is closely linked to the data warehouse maintenance. In SSIS a development framework called OVAL is used to guide the developer during the design. OVAL consists of four principles: Operations, Volume, Applications and Location.

- **Operations**: The first OVAL principle deals with identifying the operations to be performed and where in the flow of operations it is most efficient to have them. It is best to try to avoid unnecessary data conversions, and try to minimize data size by using trimmed data types for better performance.

- **Volume**: The next consideration deals with the data volume that needs to be processed. Estimations need to be done of the data volumes to be handled. The estimation is based on the size of the attributes to be extracted and the number of rows in the relevant tables that need to be transferred into the data warehouse.

- **Application**: The third principle addresses what applications and tools are best suited to keep an efficient process flow. A best practice is to avoid unnecessary merge joins in the SSIS and instead use data views in the DBMS before extracting the data. Another tip is to use ORDER BY clause in SQL statement to avoid unnecessary sort operations within SSIS.

- **Location**: The final principle deals with where the ETL process physically should run, it is necessary to know which load balances are at work and with what your ETL system has to compete.

### THE ETL PROCESS

The ETL process consists of four main steps: Extract, Clean, Conform and Deliver. These steps are overseen by operations.
Figure 11: The four ETL process steps Adapted from [Kimball et al., 2004].

In the following sections each step will be discussed in the context of objects and process control components in the SSIS environment.

**DATA EXTRACTION**

The first step in the ETL process deals with data extraction from different sources. This is done by retrieving raw data from—often a complex - diversity of sources such as relational databases, flat files and legacy systems. The extract step may include activities such as reading source-data models, connecting to and accessing data, scheduling the source system and intercepting notifications and daemons, capturing changed data and staging the extracted data to disk [Kimball et al., 2004].

In the SSIS environment extractions are managed via source adapters, called **Data Flow Sources**. These source adapters read metadata from connection managers, connect to the source, read raw data from the source, and forward it to the pipeline.

A connection manager is a logical representation of a connection that the data flow engine uses to connect physically to a source. Different types of sources need different kind of information. For example to connect to a relational database, information on database, server names, and authentication information is needed. To connect to an Excel document, information concerning filename and path needs to be stored, which data sheets are being accessed and if there are column headers or not in the document. (See figure 7)

Figure 12: The connection manager pane in BIDS

The data source adapters translate all source data into tabular data, which is data ordered in rows and columns. The extracted data is tabular independently whether it is kept in staging tables or directly into the memory buffer. Each source adapter has one output for the data flow and in many cases also an error output.
The following data flow sources are available in SSIS:

- **Excel source adapter**: This adapter is used when data is extracted from a Microsoft Excel file. You can adjust whether to read column heads and which row the table data starts at.
- **Flat File source adapter**: This adapter is used when data is extracted from a flat file, such as a text document.
- **OLE DB source adapter**: This adapter is used when data is extracted from relational databases with an OLE DB provider. You can retrieve data from tables or views or you can use SQL and stored procedures.
- **ADO NET source adapter**: This adapter is used when data is extracted from databases with a .NET provider. The DataReader Source adapter has a somewhat better performance than the OLE DB source adapter, and should be used when performance issues are of essence.
- **Raw File source adapter**: This adapter is used to pull data from an SSIS raw file. The SSIS raw file format is very efficient in reading and writing data in the SSIS environment. The Raw File source adapter is often used for data staging and for fast back up purposes.
- **XML source adapter**: This adapter is used when data is extracted from an XML source, such as a XML file or from XML held in a package variable.
- **Custom source adapter**: You can create custom adapters when none of the others apply to your needs. You can use the script transform to write a simple source adapter or you can apply any VS.NET language. This adapter enables you to retrieve data from legacy systems and from custom made systems.

### DATA CLEANSING

Data cleansing (or data scrubbing) is the technique of preprocessing data to ensure data quality. When performing analysis of data it is important that the data is accurate, complete and consistent. Data of low quality (also known as dirty data) may be the result of user entry errors, data that has been corrupted in transmission or storage, and data that has been duplicated while merging data from different sources. The data cleansing step may include activities such as enforcing column properties, enforcing structure, enforcing data and value rules, enforcing complex business rules,
building a metadata foundation to describe data quality and staging the cleaned data to disk [Kimball et al., 2004].

The general process of data cleansing is performed in three phases: Initially the error types that are present in the data needs to be defined and determined. Secondly the data set needs to be searched to identify the error instances. Thirdly, the uncovered errors need to be corrected (or the data rejected). [Maletic, 2005]

The phase **Define and Determine Error Types** deal with data profiling by establishing baseline indicators that assess data quality on data sets (rather than focusing on errors on a specific record). In this phase it is also decided what do to when these indicators are violated. This usually falls out into three possible methods: Either you fix the data, or you discard the record or you stop processing. In the SSIS environment baseline indicators and violation rules can be set up by branching data of different quality in a data flow pipeline. This can be achieved in SSIS, by using data flow transformations such as conditional splits and by applying SSIS Expression Language, a language similar to C# in the derived column transformation task. You can also use Control flow tasks like Send Mail to alert the ETL staff.

The phases **Search and Identify Error Instances** and **Correct the Uncovered Errors** both deals with data cleansing and in the SSIS environment. These two phases are handled together in the **SSIS Data Flow Pipeline** by enforcing the data quality strategy on a column-by-column and row-by-row basis through the application of transformation tasks. This can be achieved in SSIS by using data flow transformations such as fuzzy lookups that finds close or exact matches by comparing different columns and fuzzy groupings which can add values for data matching similarities. You can also use the OLE DB command transformation task to run UPDATE and INSERT SQL statements.

Another important aspect of data cleansing is to document commenced changes to the data. In SSIS this is commenced by logging that can be redirected to different sources via logging providers. This documentation might be used to evaluate the process as well as communicate knowledge back to the data providers, to improve quality earlier in the data processing lifecycle.

---

**DATA CONFORMANCE**

Data conformance is the third step in the ETL process. Conformation of data is usually necessary if the data is retrieved from different sources but also within one data source it is common with discrepancies. During this step enterprise wide agreements needs to be taken: Adherence to organizational business rules, agreeing on standards and common terminology, measures, labels, and other conformity concerns like format issues, writing notation et cetera. The result of this step is the enabling of conformed dimensions and facts. The data conformation step may include activities such as conforming business labels in dimensions, conforming business metrics and performance indicators in fact tables, deduplicating, commence householding, internationalizing and staging the conformed data to disk [Kimball et al., 2004].

To be able to integrate different data sources data conformation is essential. Data needs to be made identical in structures and code needs to be standardized. Furthermore, invalid records need to be filtered out and duplicates lifted out. Data conformation is achieved in a three step process: 1) standardizing data; 2) matching and deduplicating data; and, 3) surviving, i.e. to distilling a set of matched records into a unified set of data to build dimension table records. This can be achieved in SSIS, by using **Data Flow Transformations** such as **Merge** and **Merge Join** that combines sorted
data flows. You can also use data flow transformations such as Character Map to convert character mappings and Pivot and Unpivot transformation tasks to create columns from rows and vice versa.

Both the data cleansing step and the data conformation step are realized by transformations. The following data transformations are available in SSIS:

- **Aggregate**: This task aggregates and groups values in a data set, such as SUM and AVERAGE.
- **Audit**: Adds audit information available to the data flow.
- **Cache transform**: Populates a cache file via a cache connection manager.
- **Character Map**: This task converts data from one character mapping to another.
- **Conditional split**: This transformation task branches the pipeline depending on the conditions that are set up as baseline indicators. The conditions are specified in SSIS Expression Language, a language similar to C#.
- **Copy column**: This task creates copies of input columns to the transformation output.
- **Data Conversion**: This task converts data types, column length et cetera.
- **Data Mining Query**: Runs a data mining prediction query.
- **Derived Column**: This task applies expressions to create a new value based on the input.
- **Export Column**: Inserts data from a column to a file.
- **Fuzzy Grouping**: This transformation task finds close or exact matches in data based on one or more reference columns and then adds columns to the output with identifying matches and similarity scores.
- **Fuzzy Lookup**: This transformation task finds close or exact matches between two or more columns between the input and a row in a reference table and adds selected columns from the matched row for matching metrics.
- **Import Column**: Inserts data from a file to a column.
- **Lookup**: This transformation task joins input with columns in a reference set from a table or a cache file.

![Figure 14: The Data Flow Transformation pane in the Data Flow Toolbox in BIDS.](image)

- **Merge**: This task merges (sorted) data sets.
- **Merge Join**: This task merges data sets by using FULL, LEFT or INNER JOIN.
✓ **Multicast:** This task uses a single data set for a variety of data checks and copies all input rows to multiple outputs

✓ **OLE DB Command:** This transformation can be used to run parameterized SQL statements that updates or insert rows in a data set

✓ **Percentage Sampling:** Extracts a (sampled) subset of rows from a dataset using a percentage to specify the sample size

✓ **Pivot:** This task converts columns into rows

✓ **Row Count:** This task count rows and store this value in a variable to be used for comparison later.

✓ **Row Sampling:** Extracts a sample data set by specifying the number of rows

✓ **Script Component:** This task runs custom code to be used on the data in the pipeline

✓ **Slowly Changing Dimension:** Updates a slowly changing dimension table

✓ **Sort:** This task sorts records and remove those with duplicate values

✓ **Term Extraction:** Extracts terms from columns with text

✓ **Term Lookup:** Looks up the frequency of a term in columns with text

✓ **Union All:** This task stacks multiple data sets.

✓ **Unpivot:** This task convert rows into columns

### DATA DELIVERY

The Data delivery step concludes the process by delivering the data to the presentation area. This is done by loading the data into a multidimensional data warehouse, hence the earlier steps in the data staging all leads up to deliver data according to the structures of dimensional modeling. The data delivery step may include activities such as loading flat and snowflake dimensions, generating time dimensions, loading degenerate dimensions, loading sub dimensions, loading type 1, 2, and 3 slowly changing dimensions, conforming dimensions and conforming facts, handling late-arriving dimensions and late-arriving facts, loading multi-valued dimensions, loading ragged hierarchy dimensions, loading text facts in dimensions, running the surrogate key pipeline for fact tables, loading different fact tables, loading and updating aggregations and finally staging the delivered data to disk into a data warehouse [Kimball et al., 2004].

In SSIS, components called **Data Flow Destinations** load data into the tables in a star join schema. The following data flow destination adapters are available (See Figure 15):

✓ **ADO NET destination adapter:** This adapter loads data to a .NET provider that use a database table.

✓ **Data Mining Model Training adapter:** This adapter sends data to an Analysis Services data mining

✓ **DataReader Destination adapter:** This adapter sends data to an ADO.NET DataReader

✓ **Dimension Processing adapter:** This adapter Processes an Analysis Services dimension

✓ **Excel Destination adapter:** This adapter sends data to an Excel worksheet

✓ **Flat File destination adapter:** This adapter sends data to a flat file

✓ **OLE DB destination adapter:** This adapter sends data to an database by using an OLE DB provider

✓ **Partition Processing adapter:** This adapter processes an Analysis Services partition

✓ **Raw File destination adapter:** This adapter sends data to a raw data file

✓ **Record set destination adapter:** This adapter sends data to an ADO recordset
SQL Server Compact Destination adapter: This adapter sends data into a SQL Server Compact database
SQL Server Destination adapter: This adapter bulk inserts data into a SQL Server database
Custom destination adapter: You can also write or download custom destinations

OPERATIONS

The ETL process steps are monitored by operations such as scheduling, job execution, exception handling, recovery and restart, quality checking as well as release and support. [Kimball et al., 2004]

In SSIS data management operations are performed in the Control Flow view and in the SQL Server Management Studio. The Control flow components manage package execution. The following Control Flow Items (containers and tasks) are available in SSIS:

For Loop Container: This container repeats a task by testing a condition
Foreach Loop Container: This container repeats a task by using an enumerator
Sequence Container: This container groups multiple consecutive tasks
ActiveX Script Task: This task executes an ActiveX Script. It provides backward compatibility with Data Transformation Services (DTS), which is an earlier version of SSIS.
Analysis Services Execute DDL Task: This is an Analysis Services task that executes SQL-DDL statements that create, drop or alter cubes, dimensions and mining models in an Analysis Server
Analysis Services Processing Task: This is an Analysis Services task that processes cubes, dimensions and mining models in Analysis Services
Bulk Insert Task: This is an SQL Server task that inserts large amounts of data into a database
✓ **Data Flow Task**: This task holds data flow tasks that are used to extract data, apply transformations and load data
✓ **Data Mining Query Task**: This is an Analysis Services task that executes data mining queries in Analysis Services
✓ **Data Profiling Task**: This is a data preparation task that can be used during data profiling. It computes profiles that help you investigate a data source and identify its problems
✓ **Execute DTS 2000 Package Task**: This task executes packages built in DTS, an earlier version of SSIS
✓ **Execute Package Task**: This is a workflow task that executes an SSIS package
✓ **Execute Process Task**: This is a workflow task that runs a batch file or an application

![Control Flow Items](image)

**Figure 16**: The Control Flow Item pane in the Control Flow Toolbox in BIDS.

✓ **Execute SQL Task**: This is an SQL Server task that executes SQL or procedures
✓ **File System Task**: This is a data preparation task that creates moves or delete directories or files
✓ **FTP Task**: This is a data preparation task that download and upload files using the FTP protocol
✓ **Message Queue Task**: This is a workflow task that sends and receives messages via Microsoft Message Queueing (MSMQ)
✓ **Script Task**: This is a scripting task that extends package functionality by using scripts
✓ **Send Mail Task**: This is a workflow task that sends e-mail to notify users
✓ **Transfer Database Task**: This is an SQL Server task that transfers a database between two SQL servers
✓ **Transfer Error Messages Task**: This is an SQL Server task that transfers an error message between two SQL servers
✓ **Transfer Jobs Task**: This is an SQL Server task that transfers SQL Server agent jobs between two SQL servers
✓ **Transfer Logins Task**: This is an SQL Server task that transfers one or more logins between two SQL servers
THE DATA STAGING AREA

The staging area in the data warehouse back room does not allow data to be accessed by business queries or reports. The staging area should solely deal with data management concerning the ETL process and all business user access should be restricted to the front room presentation area. This is because the ETL process should be kept as clean and efficient as possible, without having to address presentation agreements on service levels and availability, or dealing with user access and security issues.

This section provides a brief discussion on how data and metadata can be stored in the data staging area and how the DSA can be optimized

STAGING TABLES

A common practice in the data warehouse architecture is the use of staging tables. These are temporary (or in some cases permanent) tables which raw data is loaded to from the data sources. The data is then loaded from the staging tables into the transformation process and finally delivered to the enterprise data warehouse or to several data marts.

The SSIS architecture enables data migration without using staging tables (as described in the previous section the SSIS Pipeline Architecture). However factors in the environment and business requirements might still require you to use staging tables rather than a direct “from source to destination” flow. For instance, you might not be able to extract all data at the moment when you want to process it or you might need to improve process recoverability, handle changes in data, etc. Whether to use staging tables or not is a matter of finding the right balance between physical input and output in the process. The best approach varies depending on if getting the data from the source to the destination as fast as possible is most efficient or if being able to recover from failure without having to reprocess the entire data flow is. There are several staging table patterns that are used (alone or in combinations) to address some of the most common situations. (See figure 6)
**Figure 17: Staging table patterns**

- **Staging data from Multiple Sources** is a pattern commonly used when there are multiple sources and data can't be accessed at the same time. In it the data is gathered in a staging table before it is processed. This pattern is also used to enable backup as the staging table easily can be compressed and stored elsewhere on the network.

- **Staggered Staging** is a pattern that besides storing data from (single or multiple) sources before it is processed, also keep storing the data in staging tables throughout the process. This pattern allows the staggered staging tables to act as recovery points and in case of process failure the whole process does not have to be processed again. The pattern enables faster recovery times in strained data flows but also requires storage capacity.

- **Persisted Staging** is a pattern that keeps a copy of the staging table used by the ETL process. This pattern provides opportunity to audit data outside of the current process cycle and can be used in troubleshooting the ETL process. Process inputs can also be compared with the transformed data in the staging tables throughout the process and not only between input and output from the process as a whole.

- **Accumulated Staging** is a pattern that uses a staging table to detect and accumulate data changes when the source systems do not provide a built-in mechanism to achieve this. In the accumulated staging tables data on whether new rows are inserts, updates, or deletes.

- **Chunked Accumulated Staging** is a pattern that uses the staging table to spread out the data source extraction load in smaller chunks and when all the data is gathered process them. This pattern is often used in high-transaction environments.

When implementing staging tables it is important to keep loading times as efficient as possible, only a minimum of transactions should be performed when loading data from sources into the staging area. Furthermore data is read faster from files that permits sequential reads, than from relational data.

In the SSIS environment the format that can be written to, and read from, the fastest is the **SSIS Raw File Format**. This is the recommended format for staging tables in most cases. The only properties that are needed are file path and name. As it has been optimized especially for the SSIS environment data reads are very fast. However, if you need external applications to be able to read
the files, than you should use the flat file format instead. A flat file also takes less space than an equivalent file in raw file format. This leads to that in some cases of slow disk systems flat file format can be more efficient, as less data reads against the disk system is needed.
**Tool Exercises**

In the first section, *SSIS Exercise*, you will have the opportunity to try out some of these tools yourself by following a walkthrough where you will extract data from a flat file, commit a few transformations, implement a type of Slowly Changing Dimension (SCD) technique and finally load the data into a dimension in a star join schema.

**Coverage**

The purpose of this section is to familiarize you with ETL and OLAP tools by letting you implement an exercise in Microsoft SQL Server 2008 Integration Services and Microsoft SQL Server 2008 Analysis Services.

The section Tool Exercises covers both hand on exercises as well as a general introduction to Microsoft SQL Server 2008 development and management environments.

**Preparations and Requirements**

To be able to carry out the exercises you must have access to the Data Warehousing IS5 / IV2014 portable disc containing virtualization software with Microsoft Windows operating system and MS SQL version 2008 as well as the necessary files and databases needed for the SSIS and the SSAS exercises. You can either copy the files to your hard drive or run them directly from the USB disk. Instructions for handling the virtual environment and the necessary login information are found in the course compendium.

The disks are adjusted to work on the computer environment at DSV.

**SSIS Exercise**

In this exercise you will extract data from a flat file and store it into a dimensional table. During the exercise we will look closer at problems concerning data cleaning and we will also handle changing data. After the ETL process an email alert will be send to the ETL staff.

**Create the ETL Project**

First we need to create a SSIS package. The package will be stored in a project that is part of a Visual Studio solution. To create a new project, follow these steps:

1. Open BIDS by selecting Programs → Microsoft SQL Server 2008 → SQL Server Business Intelligence Development Studio in the Windows start menu.
2. Click on the button new project… in the standard menu. (or go to File → New → Project...)
3. In the New Project window, choose the template Integration Services Project, give your project the name SSIS Exercise and choose a location to store the project file. Notice that the solution name is changed when you change the project name. (See figure 14).
4. Click the OK button.
The BIDS window will now appear and you are ready to create the first components in the control flow.

To be able to create the ETL process we will use in this exercise, you need to create at least one control flow item: The Data Flow Task. These is done by dragging and dropping the data flow task from the control flow toolbox, and drop it on the designer pane (the yellow area in the center of the BIDS application).

If the toolbox is minimized hold the cursor over the toolbox icon in the left of the BIDS window to open it. Then pin it to the background by clicking on the auto hide button in the upper right corner of the toolbox.

5. Create a Data Flow Item by dragging it from the toolbox to the designer area. (See figure 15)
6. The Data Flow Task will contain all the source, transformation and delivery tasks needed in our example ETL Process.
7. Click once on the data flow task to highlight it.
8. Click once on the text in the highlighted data flow task and rename it to Product to DW. Click somewhere in the yellow background area to view the result.

![Diagram of BIDS with highlighted data flow task]

Figure 20: Renaming items in BIDS

You have now made all the necessary preparations to begin to design the ETL process. This will be structured in the four ETL steps Data Extraction, Data Cleansing, Data Conforming and Data Delivery.

STEP 1: DATA EXTRACTION

You are now ready to start designing your ETL process in BIDS. Your first task will be to extract data from a flat file, transform it and load it into a dimensional table.

The source data file is located on the VLE homepage.

1. Download the file product.txt and save it in the IV2014 folder on the C drive of your virtual machine.

The file is called product.txt and is a semi-colon separated list of the company Elektronikkedjans products. This is a sample of it:

```
DP6847DP334296;DVD-spelare;Pioneer;dvr545HXS;6200,00;3770,00
DP6847DP334298;DVD-spelare;Philips;DVP3100V;1500,00;950,00
DP6847DP334299;DVD-spelare;Panasonic;DMR-ES35VEGS;3200,00;1970,00
DS6847DS3342101;DVD-spelare;Samsung;dvd-HR730 80GB;2800,00;1730,00
DS6847DS3342104;DVD-spelare;Samsung;DVD-R121;3000,00;1850,00
```

The first value in each row is the article number, the second value is the product category, the third value is the name of the manufacturer and the fourth consist of codes for product description. The two last values are product price and purchase price.
2. Open the Data Flow designer area by double clicking on your data flow task product to DW

Notice that the background changes from the control flow view to the Data Flow view. In the upper part of the toolbox you will see the data flow sources

3. Drag the Flat File Source item from the toolbox to the Data Flow Designer area.

You now need to create a Connection Manager that will link the source file to the data flow.

4. Double click the Flat File Source item to open the Flat File Source Editor window
5. Click on the button New... To create a new Connection Manager
6. Name the connection manager to ProductFile Manager
7. Describe the purpose of the connection manager in the description field.
8. In the Flat File Connection Manager Editor window click on the Browse... button to browse for the product.txt file
9. Browse to the IV2014 folder on the C: drive and select product.txt
10. Click Open to close the browser window
11. Check to see that the format is set to Delimited as our columns will be separated by delimiters
12. Now, click on Columns in the left pane to set the delimiters
13. Set the column delimiter to Semicolon {;}
14. Click on the button Reset Columns
15. Click on Preview in the left pane to preview the tabular data. Check to see that you have the right amount of columns (6).

We will also need to adjust the Column Width and set assign names to our columns. We will use the attribute mapping model in Figure 19 to guide us through the mapping process. Note that we don’t capture the price. These attributes are used in the fact table. In this example we will only set up a dimensional table.

![Figure 21: Attribute mapping model of the product dimension.](image)

16. Click on Advanced in the left pane to configure the column attributes according to the Dim_Produkt specification and the product attribute mapping model
17. Click OK in the Flat File Connection Manager Editor window
18. Click OK in the Flat File Source Editor window

In the connection manager area, below the designer area, is the new connection manager visible.

Now you have completed the Data Extraction step.

**STEP 2: DATA CLEANSING**

During data profiling an inconsistent use of upper and lower case was identified in the description field. We will now take care of this problem by transform all letters in lower case to upper case. This will be done by using the Character Map item in the section Data Flow Transformations in the Toolbox.
1. Drag the Character Map icon from the toolbox and drop it below the Flat File Source item in the designer area.

You now have to set the precedence constraints between the two objects so that they will be in sequence.

2. Click on the Flat File Source to highlight it.
3. Drag the green arrow from the Flat File Source and drop it on the Character Map task.

![Figure 24: Precedence constraints](image)

4. Double click on Character Map to open the Character Map Transformation Editor.

The Column named Column 3 holds the values of the product description. These values should be transformed to uppercase letters.

5. Click on the column named Column 3 in the table Available input Columns
6. Choose *in-place change* in the destination field
7. Mark Uppercase in the operation field

![Figure 25: The Character Map Transformation Editor](image)

8. Click OK in the Character Map Transformation Editor

This concludes the Data Cleansing step.

---

**STEP 3: DATA CONFIRMING**

Already in step one did we take concern for the data conformity. We extracted the data from the text file and converted it into a format that suited the destination table.
In our very simple example we only use one file as a data source. But for the sake of it, let us assume that we have two different parties that have loaded the data. One source always labels receivers as belonging to the category Förstärkare (Swedish for receiver) and another source always labels receivers as belonging to the category Receiver. When we store this information in the dimension table, we would want all rows with receivers to be labeled as Förstärkare. When our business users want to query the data they want to be able to compare receivers whether which category name has been used.

In other words, we want to replace any instances of the word Receiver with the Swedish word Förstärkare, to have one consistent language in the data warehouse. We will accomplish this by using the Derived Column transformation task.

1. Drag the Derived Column icon from the toolbox and drop it below the Character Map item in the designer area.
2. Connect the data flow from the Character Map transformation task to the Derived Column task by dragging the green arrow from Character Map and drop it on the Derived Column object.
3. Double click the Derived Column to open the Derived Column Transformation Editor.
4. In the Derived Column Name field write the name of the column where the values we want to change is stored. In this case write kategori.
5. Choose replace ‘kategori’ which will replace the old value with the new instead of adding them as a new column.

Now we will have to write an expression. First we want to find the syntax for replacing strings.

6. Click on the help button at the lower right corner of the Derived Column Transformation Editor window. (You may need to edit your preferences for online settings if it is the first time you have activated help on this installation.
7. Scroll down to the section Expressions and click on the link Functions (SSIS) to access all SSIS functions.
8. Scroll down to the link Replace (SSIS) and click on it.

Here you have a description of the function as well as its syntax and some examples. The syntax is:

\[
\text{REPLACE(character_expression, searchstring, replacementstring)}
\]

9. Write the following Expression to replace Receiver with Förstärkare:

\[
\text{REPLACE(kategori, "Receiver","Förstärkare")}
\]

10. Click on the next row. If your using the wrong syntax the text will turn red. If the text is still black, it means that the syntax is correct. The result should look something like Figure 22:
STEP 4: DATA DELIVERY

The raw data have now been extracted from the source file, cleaned and conformed and is now ready to be loaded into a multidimensional database with dimensions and fact tables. The table we are going to use is the table Dim_Product in the database ElektronikkedjaDW.

It can be a good first step to investigate the database. You can access the database via the SQL Server Management Studio (SSMS).

13. When you open SSMS you must first connect to a database engine. (See Figure 23). Click OK
14. Brose to the database *ElektronikkedjaDW* in the Object Explorer pane on the left side on the SSMS screen.

```
  Databases
    System Databases
    Database Snapshots
    elektronikkedja
      elektronikkedjaDW
        System Tables
        dbo.Dim_Butik
        dbo.Dim_Datum
        dbo.Dim_Demografi
        dbo.Dim_Kampanj
        dbo.Dim_Kund
        dbo.Dim_Produkt
          Columns
            id (PK, int, not null)
            kategori (varchar(50), null)
            tillverkare (varchar(50), null)
            beskrivning (varchar(100), null)
            artikelnr (varchar(30), not null)
          Keys
```

Figure 28: The Object Explorer pane in SSMS

15. Explore the database
16. Right click on the *elektronikkedjaDW* icon and choose New query
17. Write the following SQL statement:

```sql
USE ElektronikkedjaDW;
SELECT * FROM Dim_Product;
```

18. Click on the Execute button (marked with a red exclamation point)

As you see in the results window the database is empty as we yet haven’t loaded any data into it. (See Figure 24)

![Figure 29: An empty table viewed in SSMS](image)

19. Minimize the SSMS but do not close it.

We need to be connected to the database server in order to connect our data flow in BIDS. Go back to the BIDS environment. We will now insert a Data Flow Destination task to use it to connect to the database and store the output from our dataflow in it.

20. Select the OLE DB Destination task from the section Data Flow Destinations in the Data Flow toolbox.
21. Connect the data flow from the Derived Column transformation task to the OLE DB Destination task by dragging the green arrow from Derived Column and drop it on the OLE DB Destination
22. Double click the OLE DB Destination to open the OLE DB Destination Editor.

Your first task will be to create a connection manager to the database.

23. Click on the button New... to create open the Configure OLE DB Connection Manager window
24. Click on the button New... to create a new Connection Manager.
25. Write the name of the server you will connect to. Its name is v2003r2sql2008
26. Assure that Windows Authentication is selected in the Log on to the Server area and that
    Select or enter a database name: is selected in the Connect to a database
27. Select the database ElektronikkedjaDW from the active dropdown list (If no databases are
    shown, you probably have a connection problem to the database. Restart the SSMS and
    reconnect to the database engine
28. Click OK in the Connection Manager window
29. Click OK in the Configure OLE DBConnection Manager window
30. In the OLE DB Destination Editor check that the OLE DB connection manager is set to
    v2003r2sql2008.ElektronikkedjaDW
31. Set the data access mode to Table or view by using the dropdown list
32. Set the name of the table in the dropdown list below Name of the table or the view: to
    [dbo].[Dim_Product]

The next step is to check the mappings. Since we took the effort to name the columns already in the
data extraction step earlier, the data mapping is configured automatically. Nevertheless should you
always check these mappings.

33. Click on Mappings in the left pane and compare them to the data attribute mapping model
    in Figure 17 above
34. Click OK in the OLE DB Destination Editor
35. Check that your data flow design looks similar to the one in figure 25. Notice the new
    connection manager in the connection managers area below the designer area

Figure 30: An OLE DB Connection Manager in BIDS

You are now ready to test your dataflow. You do this by executing the data flow from the debugging
menu.

When you execute the tasks they will be yellow while they execute and either green if they succeed
or red if they fail. Errors will be shown in the error list at the bottom of the screen.

36. Go to the Debug menu and choose Start Debugging

Figure 27 depicts the data flow with two tasks completed and two tasks being executed.
Now let's go back to the SSMS window to rerun the SQL statement

37. Write the following SQL statement:

```sql
SELECT *
FROM Dim_Product;
```

38. Click on the Execute button (marked with a red exclamation point)

![Results Table]

39. To stop debugging go to the Debug menu in BIDS and choose stop debugging
You have now implemented an ETL process of loading a star join dimension using the SQL Server Integration Services.

**MANAGING CHANGES IN THE SOURCE DATA**

If you run the data flow task again 141 new rows will be added to the first one.

1. Go to the Debug menu and choose Start Debugging
2. Write the following SQL statement:

   ```
   SELECT *
   FROM Dim_Product;
   ```

3. Click on the Execute button (marked with a red exclamation point)

4. Write the following SQL statement:

   ```
   SELECT *
   FROM Dim_Product;
   ```

5. See the status bar to check that new rows have been added instead of changing the values
6. Stop debugging by going to the Debug menu in BIDS and choose stop debugging

We need to handle that the same data is loaded every time and we only want to add new rows to our dimensional tables. We also need to capture changes in old rows.

We will achieve this by applying a SCD type 2 to the dimension table Dim_Produkt:

1. Right click on the OLE DB Destination task and choose Delete. This will delete the task but notice that the OLE DB connection manager is still present in the Connection Manager pane at the bottom of the screen.
2. Drag the **Slowly Changing Dimension** icon from the toolbox and drop it below the **Derived Column** item in the designer area.
3. Connect the data flow from the **Derived Column** transformation task to the **Slowly Changing Dimension** task by dragging the green arrow from **Derived Column** and drop it on the **Slowly Changing Dimension** object
4. Double click the **Slowly Changing Dimension** item to open the **Slowly Changing Dimension Wizard**.
5. Click Next
6. Check that the `v2003r2sql2008.ElektronikkedjaDW` connection manager, that you created earlier, is shown in the connection manager dropdown list in the wizard.
7. Select `[dbo].[Dim_Produkt]` in the **Table or view** dropdown list.
7. Choose key type Business key in the `artikelnr` row

The input columns from the data flow will be shown in a table together with the columns in the Dimension. You will now have to pick out a unique identifier from the tabular data in the data flow. This key will be used to reference the input record with the Dimension record. In this case article nr is a unique identifier in the source data. This will be linked to `artikelnr` in the dimension table. (Note that `artikelnr` is not a unique identifier in the dimension table as we use ID as a surrogate key in that table.

To keep it simple we will only use SCD type 1. This will enable our dimensional table to be updated, but we will not keep any historical data.

8. Click next to go to the next step in the wizard
9. Choose `beskrivning` in Dimension Columns on the first row and then set Change Type to Changing Attribute. Do the same with `kategori` and `tilverkare`.
10. Click Next
11. Check that Fixed attributes are grayed out and that changing attributes are not marked
12. Click Next
13. Uncheck Enable inferred member support.
14. Click Next.
15. Click Finish
Your slowly changing dimension has now been added, two new tasks have also appeared on your designer area. The Insert destination is an OLE DB destination equivalent to the one you did yourself in the previous section. The OLE DB Command task runs the following statement to update the dimensional table:

```
UPDATE [dbo].[Dim_Produkt]
WHERE [artikelnr] = ?
```

Now, let us empty the test database elektronikkedjaDW and then run the process to see how it works.

16. If SSMS is closed restart it and then create a new query by clicking on the new query button in the upper left corner of the SSMS screen.
17. Write the following statement to empty the database (double check that you have written elektronikkedjaDW and not elektronikkedja as all data from the table will be lost when you run these statements)

```
USE elektronikkedjaDW;
DELETE FROM Dim_Produkt;
SELECT *
FROM Dim_Produkt;
```

18. Click on the button Execute.

There should be no records in the result pane.

19. Go back to the BIDS environment
20. Execute the data flow by selecting Start Debugging from the Debug menu
21. Go back to SSMS and replace the previous statements with the following:
SELECT *
FROM Dim_Produkt;

22. Click on the button Execute.

You should have 141 rows in the result set

![Database Query Result]

Figure 35: The end result

23. Now open the file Produkt.txt in the IV2014 folder and replace the value “Canon” on the first four rows, with the value “Luxor”

24. Go back to SSMS and run the statement with the following:

    SELECT *
    FROM Dim_Produkt;

25. Click on the button Execute.

You should still have 141 rows but the value “Canon” is replaced with the value “Luxor”
This document is primarily based on [Mundy et al., 2006], [Turley et al., 2007], [Kimball et al., 2004].

REFERENCE LITERATURE

- [Payne, 2006] Payne, Business Intelligence and Data Warehousing in SQL Server 2005 Microsoft (2006),