



IBM Infosphere DataStage

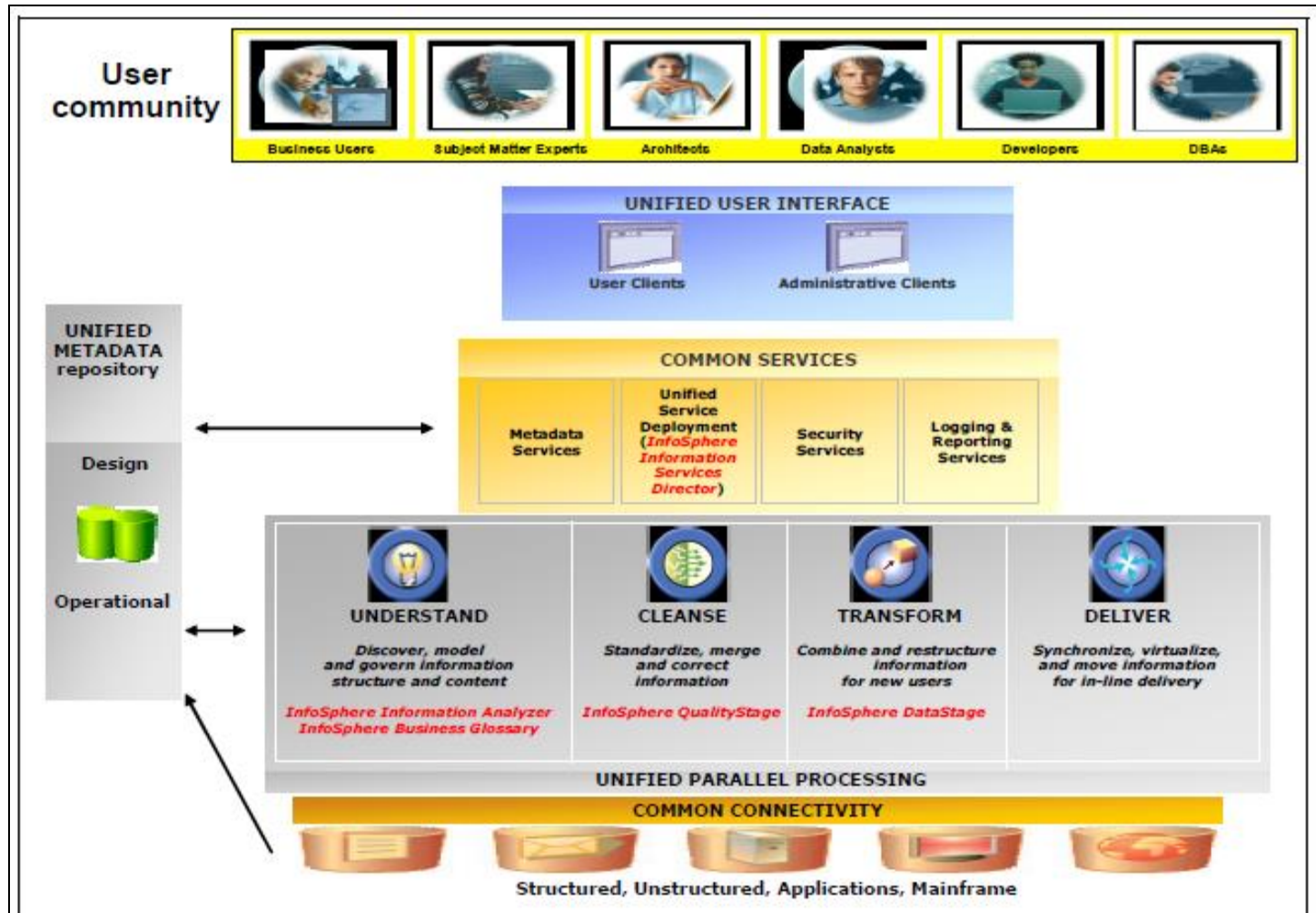
Presented by : Prashant Tikone

This presentation is the intellectual property of Cybage Software Pvt. Ltd. and is meant for the usage of the intended Cybage employee/s for training purpose only. This should not be used for any other purpose or reproduced in any other form without written permission and consent of the concerned authorities.

Agenda

Introduction to Infosphere Platform
DataStage Concepts
Installation and Setting up
DataStage Stages
Demos / Videos

Infosphere Platform



Infosphere Platform

IBM InfoSphere DataStage

Enables organizations to design data flows that **extract** information from multiple source systems, **transform** it in ways that make it more valuable, and then **deliver** it to one or more target databases or applications.

IBM InfoSphere QualityStage

Designed to help organizations **understand and improve** the overall quality of their data assets, IBM InfoSphere QualityStage provides advanced features to help investigate, repair, consolidate, and validate heterogeneous data within an **integration workflow**.

IBM InfoSphere Information Services Director

IBM Information Server provides a unified mechanism for **publishing and managing shared Service Oriented Architecture (SOA) services** across data quality, data transformation, and federation functions, allowing information specialists to easily deploy services for any information integration task and consistently manage them.

Infosphere Platform

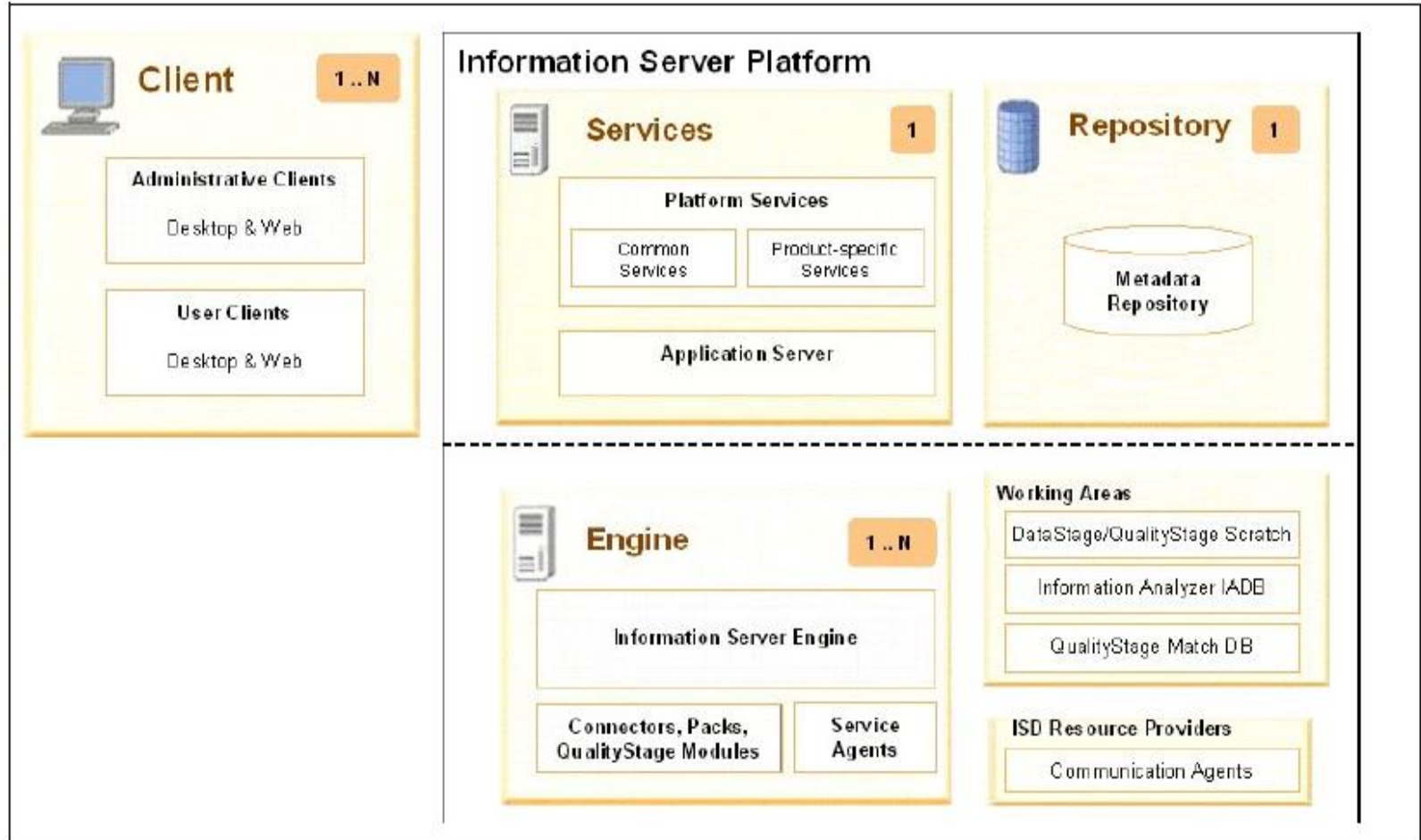
IBM InfoSphere Information Analyzer –

IBM InfoSphere Information Analyzer profiles and analyzes data so that you can deliver trusted information to your users. It can automatically **scan** samples of your data to determine their **quality and structure**.

IBM InfoSphere Business Glossary

IBM Information Server provides a Web-based tool that enables business analysts and subject-matter experts to create, manage, and share a common **enterprise vocabulary and classification system**.

Information Server Architecture



Information Server architecture

Client tier

IBM Information Server provides a number of client interfaces, optimized to different user roles within an organization. The clients tier includes

- IBM InfoSphere DataStage and
- IBM InfoSphere QualityStage
- IBM Information Server console
- IBM Information Server Web console.

Server tiers

The server tiers of the Information Server Platform that includes

- Information Server Engine
- Platform Services
- Repository,
- Working Areas,
- Information Services Director Resource

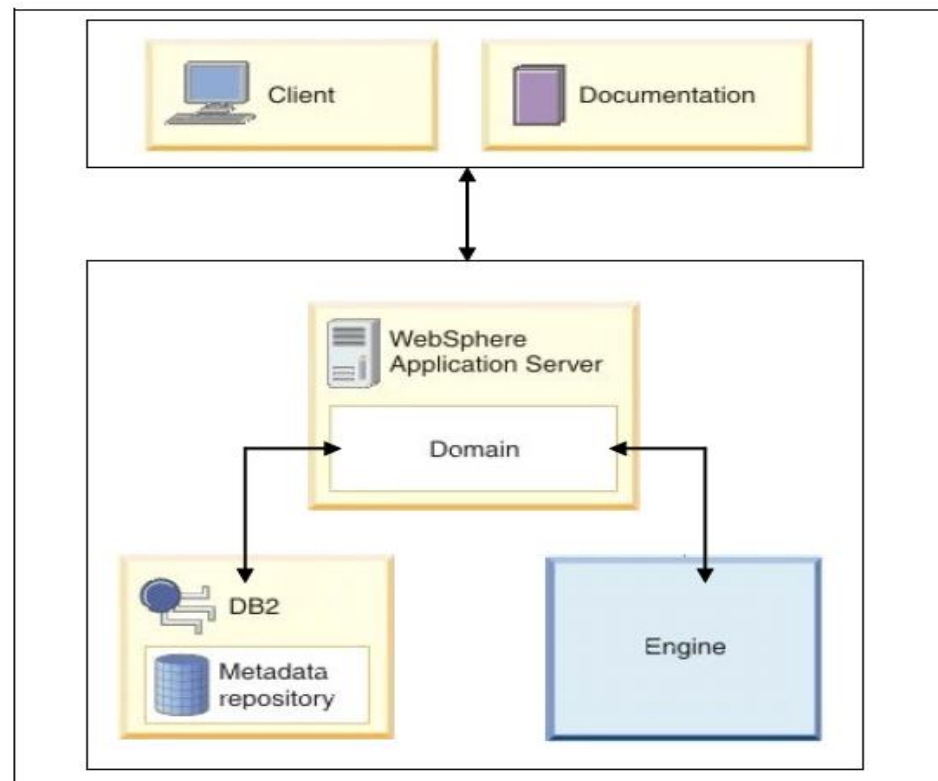
Topologies

IBM Information Server supports multiple topologies to satisfy a variety of your data integration and hardware business requirements.

- Two-tier
- Three-tier
- Cluster
- Grid

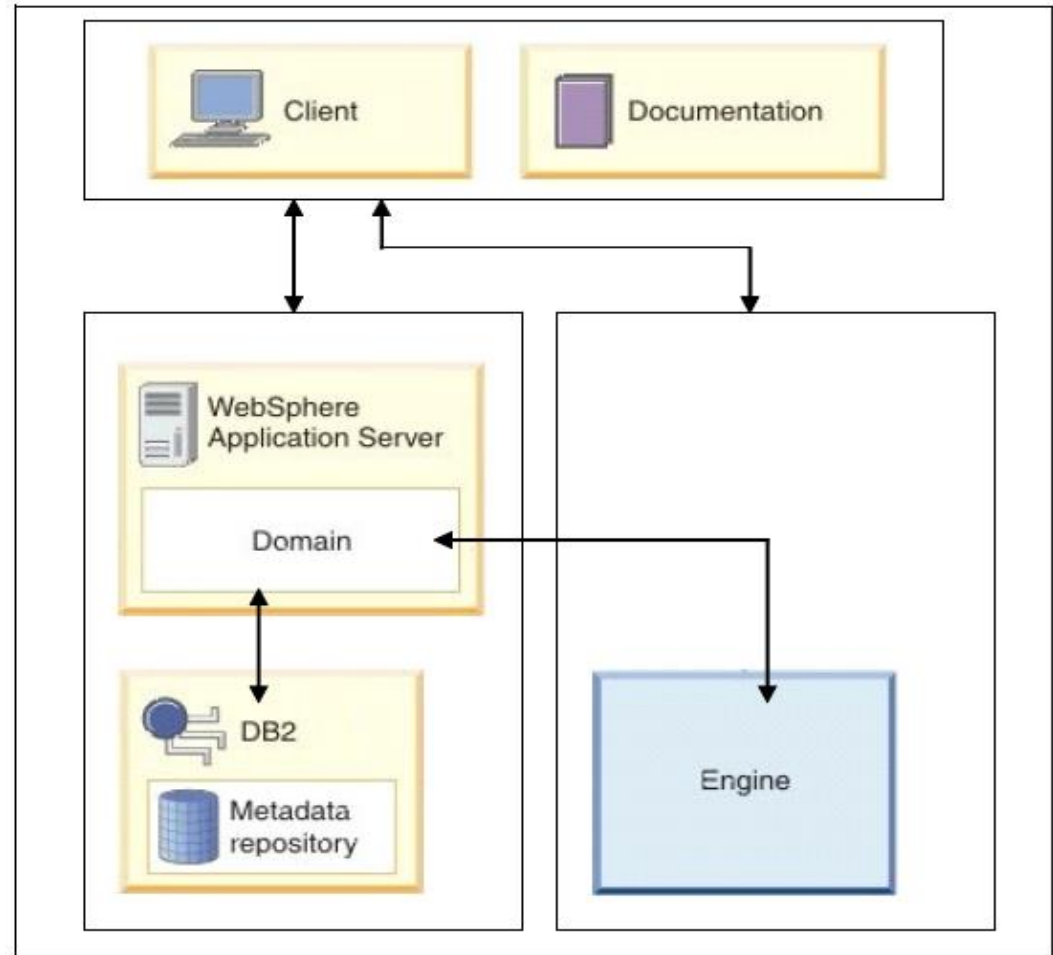
Topologies

Two-tier - The engine, application server, and metadata repository are all on the same computer system, while the clients are on a different machine



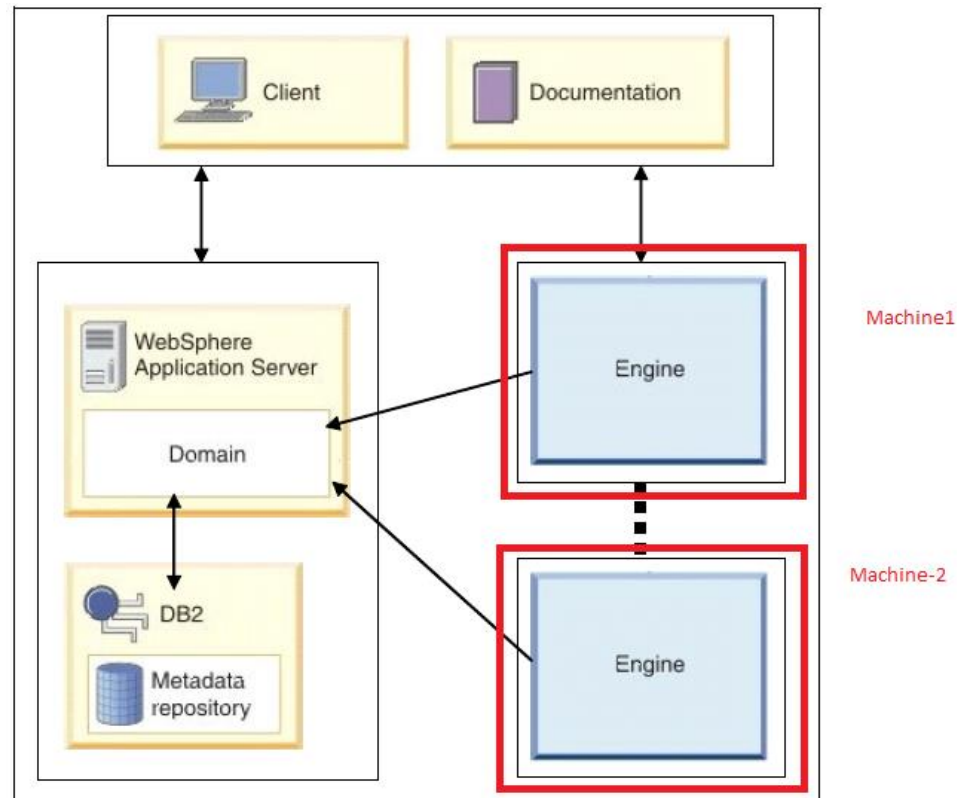
Topologies

Three-tier -
The engine is on one machine, the application server and metadata repository are co-located on another machine, while the clients are on a third machine as



Topologies

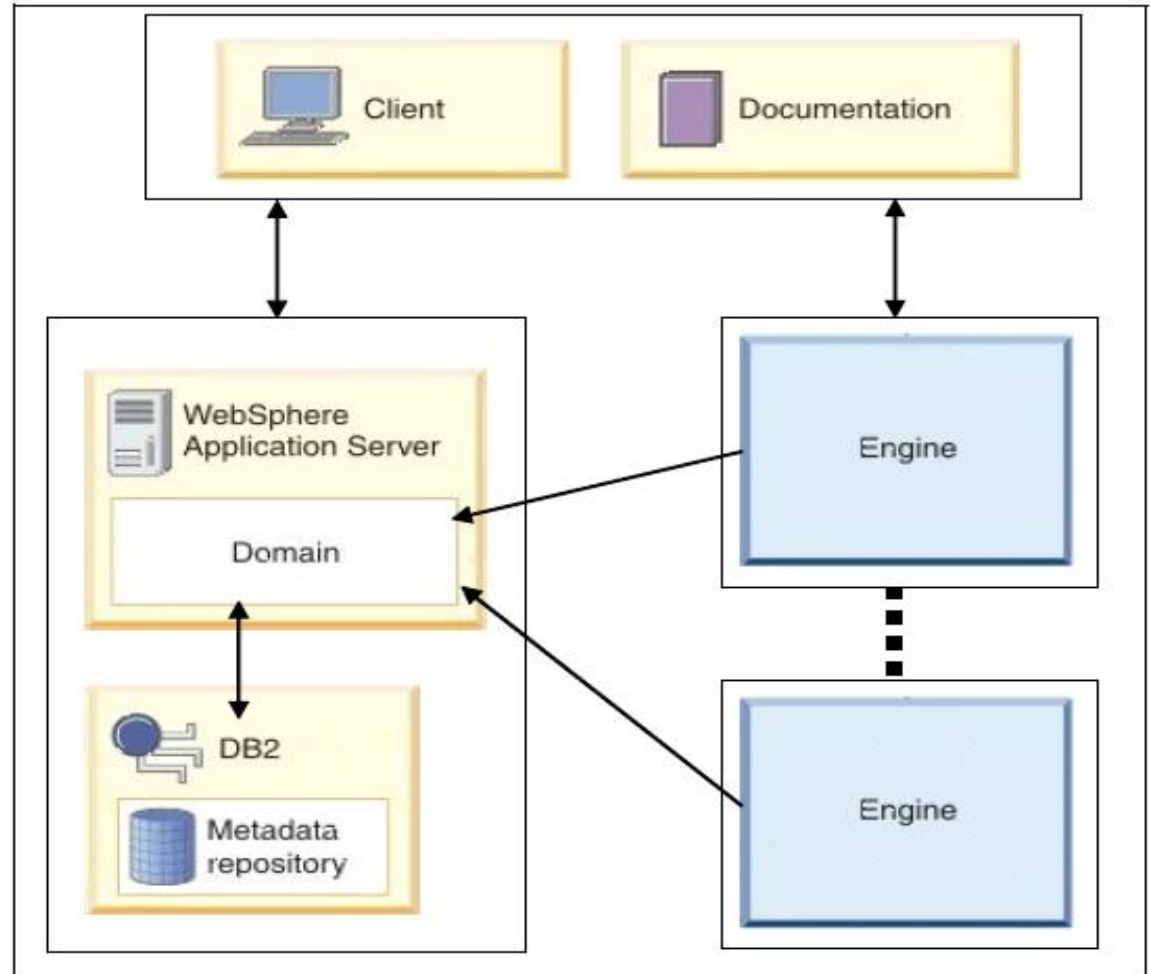
Grid - Grid topology is very similar to that of a cluster topology, the execution engines are distributed over multiple machines. The machines over which a job executes are dynamically determined



Topologies

Cluster
Variation of
the three-
tier
topology
with the
engine
duplicated
over
multiple
computers.

Single parallel
job
execution
can span
multiple
computers



DataStage Concepts

Core capabilities:

- Connectivity to a wide range of mainframe, legacy, and enterprise applications, databases, file formats, and external information sources.
- Prebuilt library of more than 300 functions including data validation rules and very complex transformations.
- Maximum throughput using a parallel, high-performance processing architecture.
- Enterprise-class capabilities for development, deployment, maintenance, and high-availability. It leverages metadata for analysis and maintenance. It also operates in batch, real time, or as a Web service.

DataStage Concepts

Data transformation

The process by which source data is selected, converted, and mapped to the format required by targeted systems.

Manipulates data to bring it into compliance with business, domain, and integrity rules and with other data in the target environment.

DataStage Concepts

Transformations Types

Aggregation - Consolidating or summarizing data values into a single value.

Conversion - Ensuring that data types are correctly converted

Cleansing - Resolving inconsistencies and fixing the anomalies in source data.

Derivation - Transforming data from multiple sources by using a complex business rule or algorithm.

Enrichment - Combining data from internal or external sources to provide additional meaning to the data.

Normalizing - Reducing the amount of redundant and potentially duplicated data.

Combining - The process of combining data from multiple sources via parallel Lookup, Join, or Merge operations.






Pivoting - Converting records in an input stream to many records in the appropriate table in the data warehouse or data mart.

Sorting - Grouping related records and sequencing data based on data or string values.

DataStage Concepts

Jobs –

Consists of individual stages linked together which describe the flow of data from a data source to a data target.

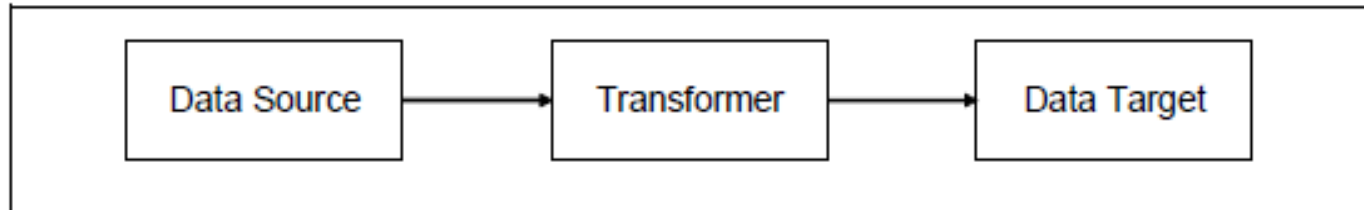
Icon	Stage	Description
	Transformer stage	Performs any required conversions on an input data set, and then passes the data to another processing stage or to a stage that writes data to a target database or file.
	Sort stage	Performs complex high-speed sort operations.
	Aggregator stage	Classifies data rows from a single input data set into groups and computes totals or aggregations for each group.
	Complex Flat File stage	Extracts data from a flat file containing complex data structures, such as arrays or groups.
	DB2 stage	Reads data from or writes data to IBM DB2.

DataStage Concepts

Parallel processing

In a parallel job, you can have multiple instances of each process to run on the available processors in your system.

Example - simplest jobs shown below has a data source, a Transformer (conversion) stage, and the data target. The links between the stages represent the flow of data into or out of a stage.



Types of parallel processing

- Pipeline
- Partitioning.
- Hybrid

DataStage Concepts

Pipeline parallelism

- All stages run concurrently, even in a single-node configuration.
- As data is read from the Oracle source, it is passed to the Transformer stage for transformation,
- Instead of waiting for all source data to be read, as soon as the source data stream starts to produce rows, these are passed to the subsequent stages.
- The Information Server Engine always executes jobs with pipeline parallelism.

DataStage Concepts

Partition parallelism

- Partitioning the data into a number of separate sets, with each partition being handled by a separate instance of the job stages.
- Partition parallelism is accomplished at runtime, instead of a manual process that would be required by traditional systems.
- Specify the algorithm to partition the data, not the degree of parallelism or where the job will execute.
- Using partition parallelism the same job would effectively be run simultaneously by several processors, each handling a separate subset of the total data.
- At the end of the job the data partitions can be collected back together again and written to a single data source.

DataStage Concepts

Hybrid –

Information Server engine **combines pipeline and partition** parallel processing to achieve even greater performance gains.

Stages processing partitioned data and filling pipelines so the next one could start on that partition before the previous one had finished.

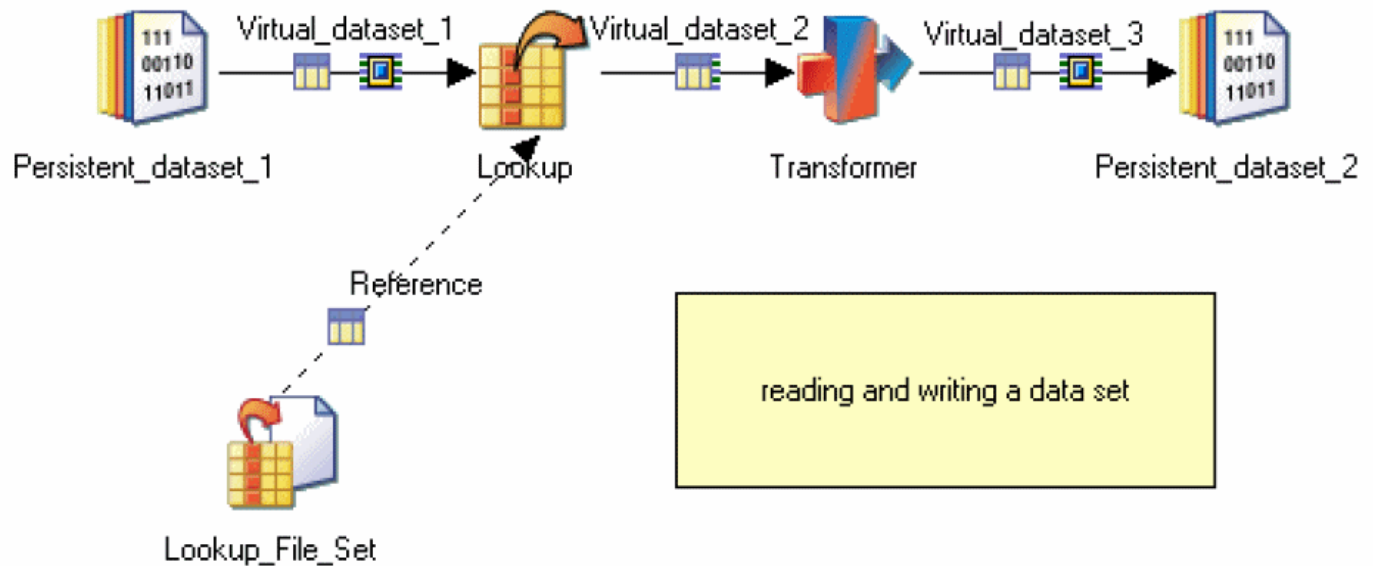
DataStage Stages

Aggregator stage – It is a processing stage which converts data rows from a single input link into groups and computes totals or other aggregate functions for each group. The summed totals for each group are output from the stage via an output link.



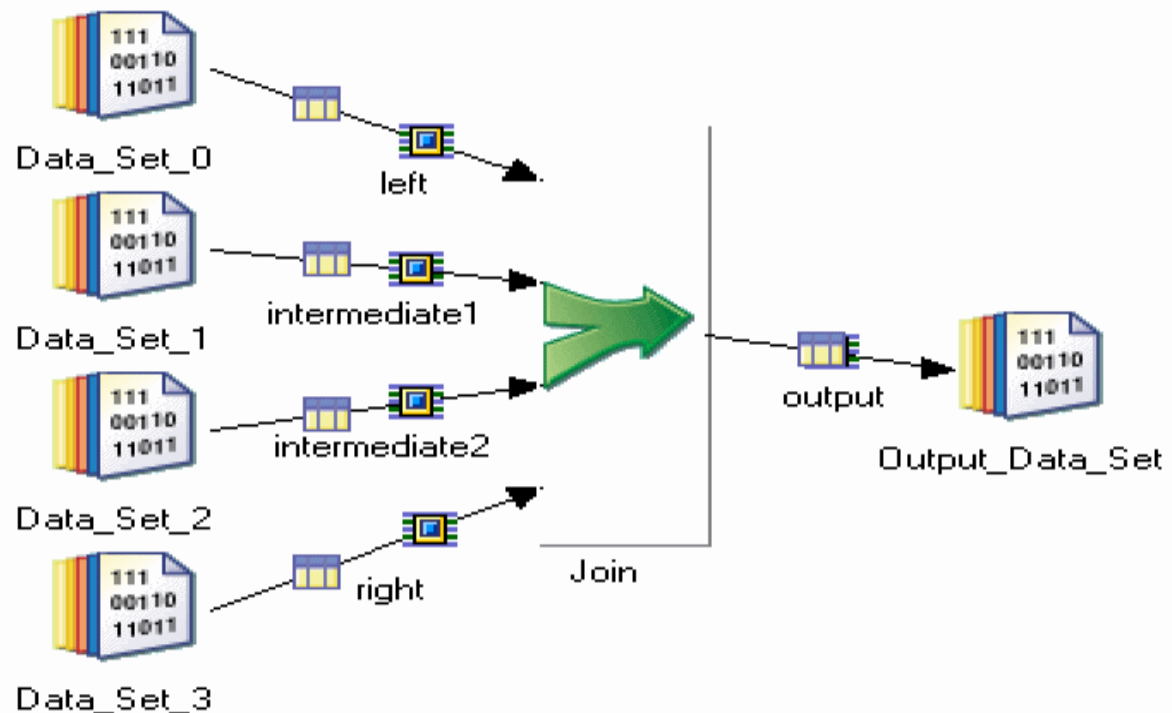
DataStage Stages

Data Set stage - It allows you to read data from or write data to a data set. The stage can have a single input link or a single output link as shown



DataStage Stages

Join stage – It is a processing stage. It performs join operations on two or more inputs to the stage and then outputs the resulting data set.



DataStage Stages

Join stage can perform one of four join operations

“Inner” transfers records from input data sets whose key columns contain equal values to the output data set.

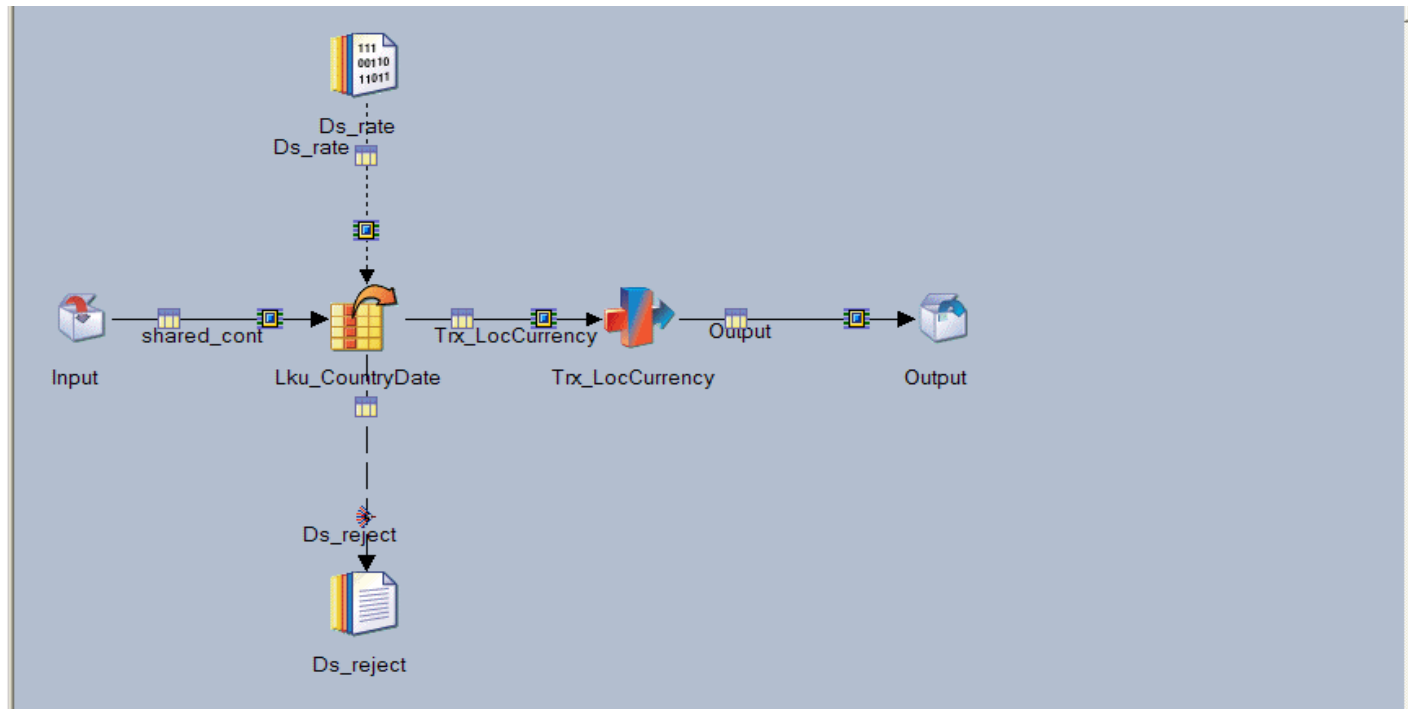
“Left outer” transfers all values from the left data set but transfers values from the right data set and intermediate data sets only where key columns match.

“Right outer” transfers all values from the right data set and transfers values from the left data set and intermediate data sets only where key columns match.

“Full outer” transfers records in which the contents of the key columns are equal from the left and right input data sets to the output data set. It also transfers records whose key columns contain unequal values from both input data sets to the output data set.

DataStage Stages

Lookup stage - It is used to perform lookup operations on a data set read into memory from any other Parallel job stage that can output data. It can also perform lookups directly in all DBMSs or in a lookup table contained in a Lookup File Set stage.



DataStage Stages

Diagram show the mapping of one column each from the two input links (shared_cont and Ds_rate) to the output link Trx_LocCurrency. The column definitions of each of these links is shown in the bottom pane.

Lku_CountryDate - Lookup Stage

The diagram illustrates the mapping of columns from two input links to the output link Trx_LocCurrency in a Lookup Stage.

Input Links:

- shared_cont:**

Key Expression	Range	Column Name
	<input type="checkbox"/>	COUNTRY
	<input type="checkbox"/>	TOTAL_US
	<input type="checkbox"/>	LookupDa
- Ds_rate:**

Key Expression	Key Ty	Column Name
shared_cont.COUNTRY_ISO_CODE	=	country_iso_cod
shared_cont.LookupDate	=	date
		rate_from_usd

Output Link: Trx_LocCurrency

Derivation	Column Name
shared_cont.TOTAL_USD	TOTAL_USD
Ds_rate.rate_from_usd	RATE_FROM_US

Column Definitions:

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
1	COUNTRY_ISO_CODE	<input checked="" type="checkbox"/>	Char	Unicode	3		Yes	<none>
2	TOTAL_USD	<input type="checkbox"/>	Decimal		10	2	Yes	<none>
3	LookupDate	<input checked="" type="checkbox"/>	Date				No	

Trx_LocCurrency Column Definitions:

	Column Name
1	TOTAL_U
2	RATE_FR

DataStage Stages

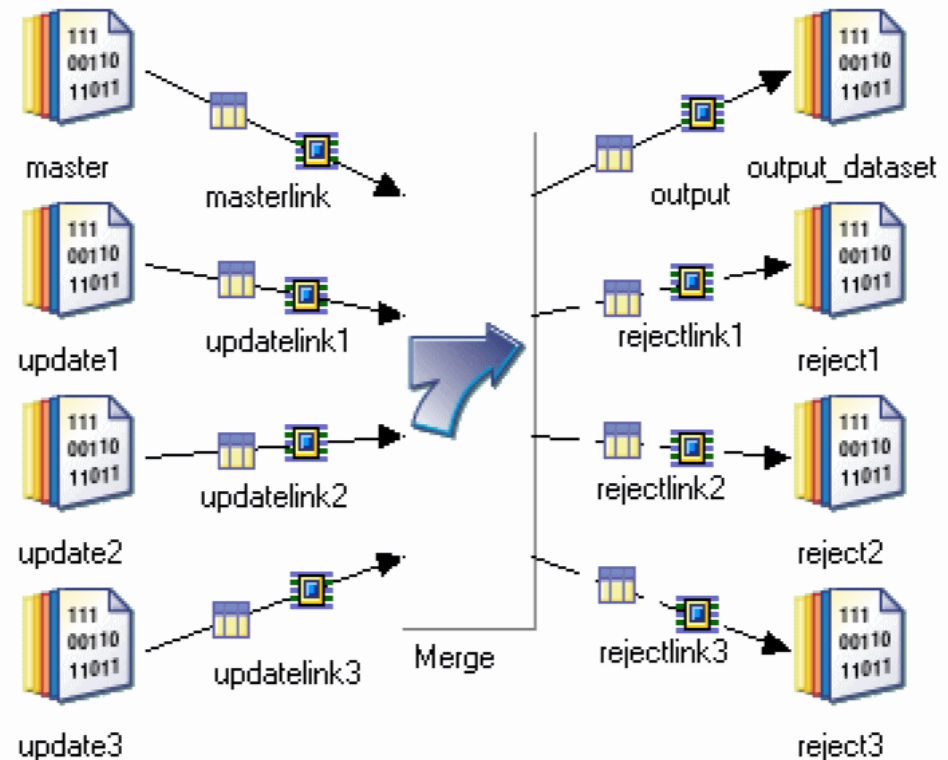
Each record of the output data set contains columns from a source record plus columns from all the corresponding lookup records where corresponding source and lookup records have the same value for the lookup key columns.

The optional **Reject link** carries source records that do not have a corresponding entry in the input lookup tables.

Range lookup – It compares the value of a source column to a range of values between two lookup table columns. If the source column value falls within the required range, a row is passed to the output link.

DataStage Stages

Merge stage - It can have any number (more than 1) of input links, a single output link, and the same number of reject links as there are update input links.



DataStage Stages

Merge stage **combines a master data set with one or more update data sets**. The columns from the records in the master and update data sets are merged so that the output record contains all the columns from the master record plus any additional columns from each update record that are required.

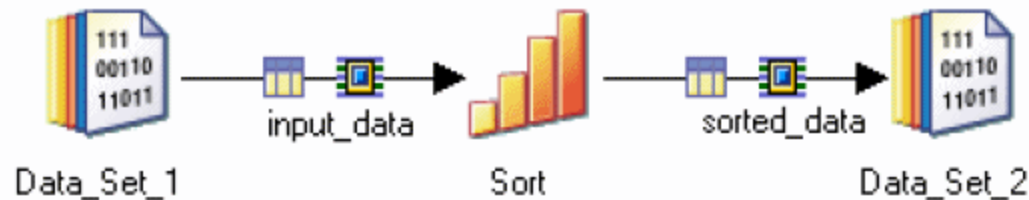
A master record and an update record are merged only if both of them have the **same values for the merge key column(s)** that you specify. Merge key columns are one or more columns that exist in both the master and update records.

The data sets input to the Merge stage must be **key partitioned and sorted**. This ensures that rows with the same key column values are located in the same partition and will be processed by the same node

DataStage Stages

Sort stage - It is used to perform more complex sort operations than can be provided for on the Input page Partitioning tab of parallel job stage editors.

The Sort stage has a single input link that carries the data to be sorted, and a single output link carrying the sorted data. We need to specify sorting keys as the criteria on which to perform the sort.



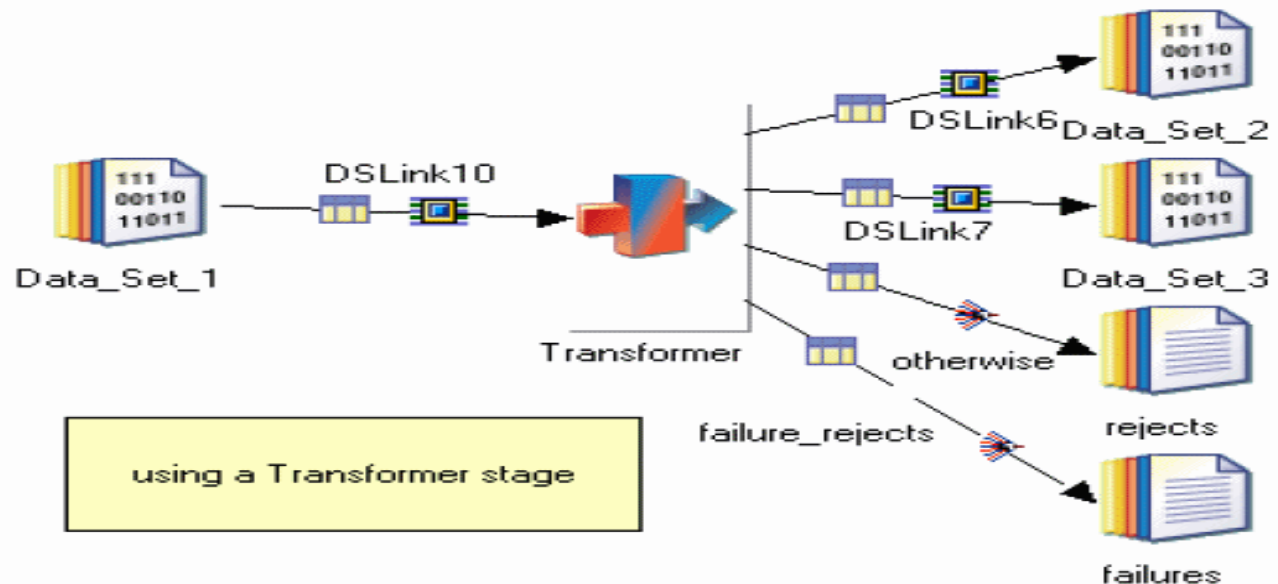
DataStage Stages

Transformer stage – It allow you to create transformations to apply to your data. These transformations can be simple or complex and can be applied to individual columns in your data.

Transformations are specified using a powerful set of functions such as date & time, logical, mathematical, null handling, number, raw, string, vector, type conversions, type casting, and utility functions.

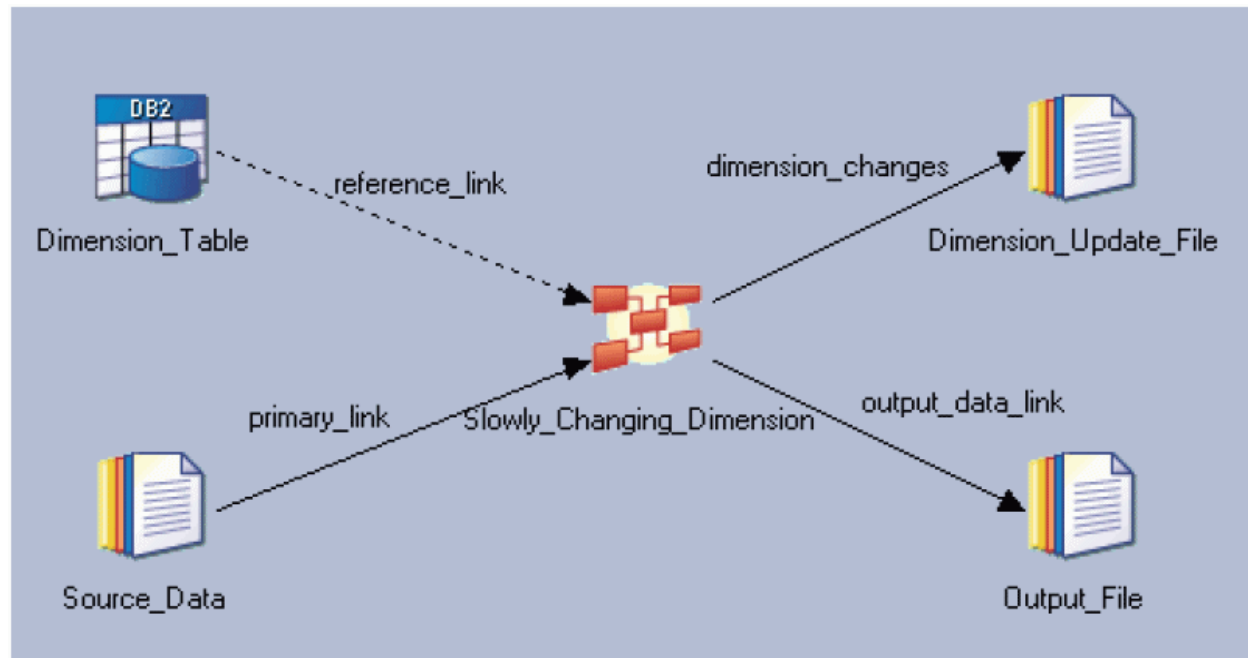
DataStage Stages

Transformer stages can have a single input and any number of outputs. It can also have a reject link, which takes any rows that have not been written to any of the outputs links by reason of a write failure or expression evaluation failure



DataStage Stages

Slowly Changing Dimension (SCD) stage is a processing stage that works within the context of a star schema database. The SCD stage has a single input link, a single output link, a dimension reference link, and a dimension update link



DataStage Stages

SCD stages support both SCD Type 1 and SCD Type 2 processing.

SCD Type 1 - Overwrites an attribute in a dimension table.

SCD Type 2 - Updates the existing row to indicate it expired and adds a new row to the dimension table.

Each SCD stage processes a single dimension and performs lookups by using an equality matching technique. If the dimension is a database table, the stage reads the database to build an in memory lookup table of all the current dimension entries.

If a **match is found**, the SCD stage updates rows in the dimension table to reflect the changed data.

If a **match is not found**, the stage creates a new row in the dimension table. All of the columns that are needed to create a new dimension row *must be* present in the source data.

Demo Links

Introduction

<https://www.youtube.com/watch?v=arUngmXLXmk>

Installation

<https://www.youtube.com/watch?v=-zeTbpVp2IM>

DataStage Parallel Job

<https://www.youtube.com/watch?v=i2wDEnODDbI>

<https://www.youtube.com/watch?v=iAP8XeGBseg>

<https://www.youtube.com/watch?v=2JG4qfLUwQ8>

DataStage Monitoring

https://www.youtube.com/watch?v=qOI_6HqyVes

Question?





Thank You