

## **Ab Initio Interview Questions**

### **Q. What do you understand by Ab Initio? / Give a brief introduction of Abinitio.**

Ab Initio, also known as Abinitio, is a tool used to extract, transform, and load data. 'Abinitio' is a Latin word that means 'from the beginning'.

It was named Abinitio because Sheryl Handler and their team started it after the bankruptcy of their previous company.

Sheryl Handler was the former CEO of Thinking Machines Corporation, and he decided to start this company as a new beginning when the Thinking Machines Corporation went bankrupt.

It is mainly used for data analysis, data manipulation, batch processing, and graphical user interface (GUI) based parallel processing for businesses.

### **Q. What is Ab Initio Software?**

Ab Initio Software is an American multinational private enterprise software corporation headquartered in Lexington, Massachusetts.

Ab Initio Software specializes in high-volume data processing applications and enterprise application integration.

The Ab Initio software provides several products on a platform for parallel data processing applications.

### **Q. What is Abinitio?**

“Abinitio” is a latin word meaning “from the beginning.”

Abinitio is a tool used to extract, transform, and load data.

It is also used for data analysis, data manipulation, batch processing, and graphical user interface based parallel processing.

Ab Initio is popular software that offers data processing applications and enterprise application integration. It provides a single platform for data analysis, complex events, batch processing, data manipulation, quantitative, and qualitative data processing.

Ab Initio software helps you build large-scale data processing applications and run them in parallel environments.

### **Applications of Ab initio**

- It is used for application development and execution environments.
- It is a tool used in most of the big industries like insurance, banking, logistics, stock market, retail, finance, telecommunication to process complex and enormous volumes of data which gives stability and reliability.
- Ab initio solves the most challenging data processing issues for many of the leading organizations in many fields like finance and insurance.
- ETL for data warehouses, data marts and some operational data sources.
- It is also used for the parallel data cleansing and also for the validation.
- It can be seen in the parallel data transformation and also in the filtering.
- Ab initio ETL tool can also be seen in the high-performance analytics.
- It can be seen in real time and also in the parallel data capture.

- ETL for data warehouses, data marts and operational data sources.

- Parallel data cleansing and validation

- Parallel data transformation and filtering

- High Performance analytics

- Real time, parallel data capture

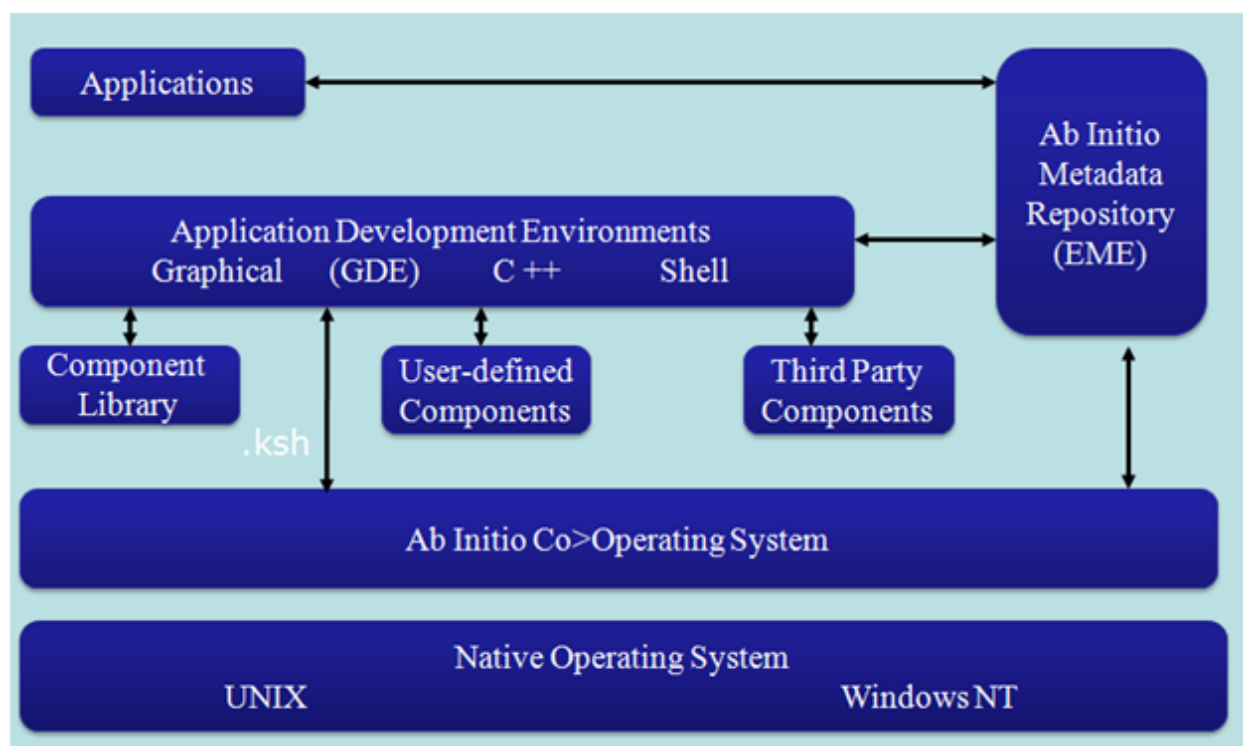
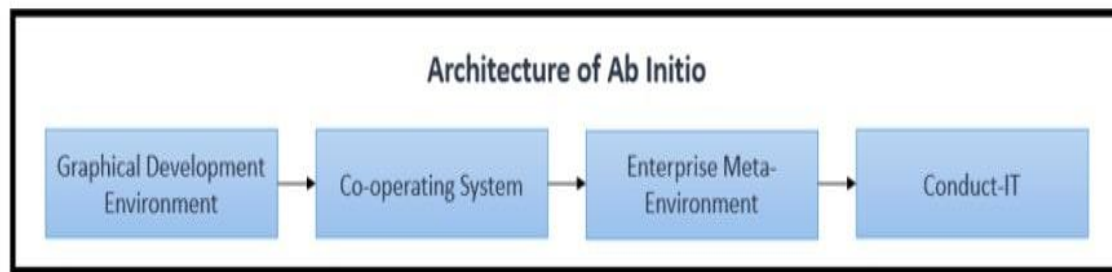
**Advantages of Abinitio**

- large amount of data can be handled.
- Development time is less.
- more parallelism (3).
- Scalability (utilize the resources).
- Version control (EME).
- Enterprise management's.

**Q. Which industries mainly used Abinitio?**

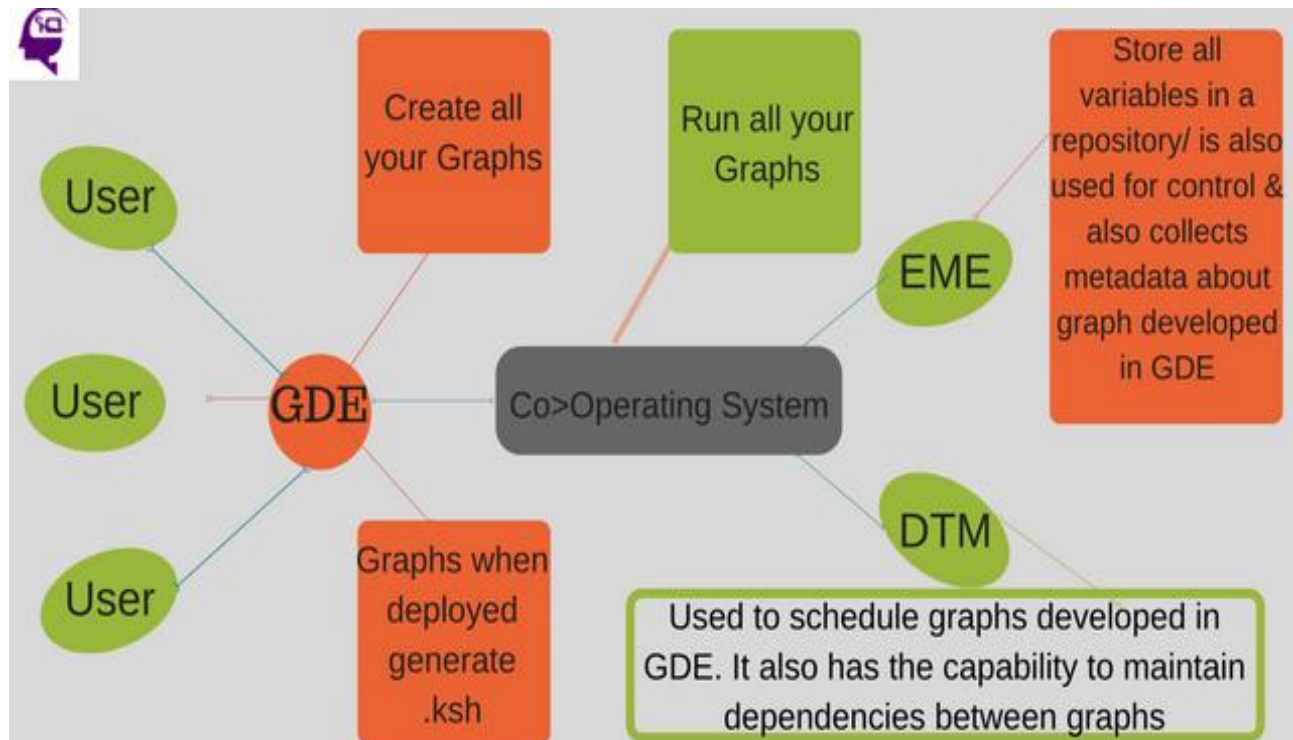
Abinitio Software applications are most widely used in Business Intelligence Data Processing Platforms to build most business applications such as operational systems, distributed application integration, complex event processing to data warehousing, and data quality management systems.

**Q. What is the architecture Of Ab Initio?**



### **Ab Initio ETL tool Architecture**

- Ab-Initio program works with the client-server model.
- The client is referred to as "Graphical development environment" (GDE).
- The server is called "Co-Operating System".
- The Co-working system can dwell on a mainframe or UNIX machine.
- The Ab-Initio code is referred to as graph, which has received .mp extension.
- The graph from GDE is required to be deployed in corresponding .Ksh version.
- In a Co-operating approach the corresponding .Ksh in a run to do the required job.



**The Ab Initio is a business intelligence software containing 6 data processing products:**

1. Co>Operating system (Co>Op v2.14, 2.15..)
2. The Component Library
3. Graphical Development Environment (GDE v1.14, 1.15...)
4. Enterprise Meta>Environment (EME v3.0...)
5. Data Profiler
6. Conduct>IT

**GDE & Co-operating system is a Client-Server Architectures**

## **1. Co>Operating System:**

Co>Operating system is an engine for you graph to Run. It is not visible, it installed on your OS.

Co>Operating system is nothing like distributed operating system which runs as a back end for GDE.

Co>operating system is a server for Abinitio.

The Co>Operating System® product is a platform for building and deploying large, sophisticated business data processing applications.

These applications can reliably process large amounts of data by running scalably on many CPUs within a single server and across networks of servers or containers (computing clouds).

The Co>Operating System product provides complete portability of applications across Unix, Linux, Hadoop, Windows, and mainframe (z/OS) servers.

Co>Operating System, which your system administrator installs on a host UNIX or Windows NT Server, as well as on processing nodes. (The host is also referred to as the control node.)

Co>Operating System **operates on the top of the operating system and works as a base for all Ab Initio processes**. It can run on operating systems like Windows, Linux, Solaris, AIX, HP-UX, and z/OS. It provides Ab Initio extensions through which ETL processes can be controlled.

Ab Initio Co>operating system is a foundation for all Ab Initio functions and provides a base for all Ab Initio processes. It runs on a variety of process environments like AIX, HP-UX, Solaris, Linux, z/OS, and windows.

### **It presents following points:**

- Control and run Ab Initio graphs and controls the ETL methods
- Provides Ab Initio extensions to the operating system
- ETL processes monitoring and debugging
- Metadata administration by interacting with EME

**Ab initio Co>Operating system** is the foundation for all the Ab initio applications. It provides a general engine for the integration of all kinds of data processing and communication between all the ETL tools within the given platform.

It runs on Windows, Linux, Unix, OS/390, zOS on the mainframe. It enables distributed and parallel execution, platform independent data transport, establishing checkpoints, and process monitoring.

It is the foundation for all the Ab initio functions and provides a base for all the Ab initio processes. It controls and runs the Ab initio graphs and controls all of the ETL methods.

It provides the extensions to the operating systems powered by Ab initio. It also helps for debugging the information.

**Co>Operating System** is installed on the operating system.

CoOperating System It operates on top of the operating system, and the ab initio provides this and it the base for all Ab Initio processes.

Air commands are one of the features that can be installed on different operating systems like UNIX, Linux, IBM etc.

This co-op is installed on particular o.s platform that is called native o.s .coming to the eme, it's just as a repository in Informatica, its hold the metadata, transformations, dbconfig files source and targets information.

**These are the following features that it provides,**

- Manages and runs Ab Initio graphs and control the ETL processes
- Providing the extensions
- ETL processes monitoring and debugging
- Metadata management and interaction with the EME

## **2. Graphical Development Environment:**

GDE is a graphical application for developers which is used for designing and running Ab Initio graphs.

You are going to create a business application or Graph.

- Graph is a combination of dataset component and program.
- Component connected by a flow.

The Graphical Development Environment™ (GDE®) product allows for extremely rapid development of complex and enterprise-wide applications. The graphical nature of the product encourages a clearer understanding of these applications by large and diverse groups of people.

**GDE** It's a designing component and used to run the ab initio graphs.

Graphs are formed by the components (predefined or user-defined) and flows and the parameters. It provides the ETL process in Ab Initio that is represented by graphs.

Ability to run, debug the process logs jobs and trace execution logs

GDE provides an intuitive graphical interface for enhancing and executing applications. Possible without any difficulty drag-and-drop add-ons from the library on to a canvas, configure them and fasten them into flowcharts. Ab Initio GDC graph compilation system outcome within the new release of a UNIX shell script which may be executed on a machine where GDE is not installed.

**Graphical Development Environment (GDE)**, which you install on your PC

(Client node) and configure to communicate with the host (control node).

### **Graphical Development Environment (GDE)**

Is the graphical application for designing and running Ab Initio graphs.

Users can access the EME metadata through the GDE web browser or Co>Operating system command line.

Coming to gde its is an end-user environment where we can develop the graphs (mapping just like in Informatica) designer uses the gde and designs the graphs and save to the eme or sandbox it is at the user side. Where eme is at the server-side.



### **Graphical Development Environment (GDE):**

Ab initio Graphical Development Environment (GDE) gives an intuitive graphical interface for editing and executing the applications. It is very easy to drag components from the library onto a canvas, to configure them, and connect them via flowcharts. Here, the code development tool is developer friendly.

### **3. Enterprise Meta>Environment:**

EME is repository and environment to retailer and managing metadata. It has the potential to retailer both business and technical metadata.

EME metadata can also be accessed from the Ab Initio GDE, internet browser or Ab Initio Co>Op command line.

The Enterprise Meta>Environment® (EME®) product is an enterprise data catalog and metadata management solution. It provides broad capabilities for all metadata created and used by the Co>Operating System product and a large number of third-party products. The EME product versions and stores schema-level metadata, business metadata, data transformation rules, applications (graphs), and operational statistics. The EME product uses technical metadata to automatically derive dependency and lineage information. It also includes a set of connectors for integrating information from a wide range of third-party design and reporting tools.

Ab initio Enterprise Meta>Environment is a data store that allows tracking changes in the developed graphs and metadata used in their development. It also offers some tools such as independent analysis, metadata management, statistical analysis, and version controlling. It has the potential to retailer both the business and technical metadata. It can also be accessed from the Ab initio Graphical Development Environment (GDE) or from any internet browser.

### **4. Component Library:**

The Ab Initio Aspect Library is a reusable program module for sorting, data transformation, and excessive-space database loading and unloading.

The Ab initio Component Library is a reusable software module for sorting, data transformation, and high-speed database loading and also unloading. It is a flexible and extensible tool which adapts at the runtime to the old formats of the given data entered.

The Ab initio Component Library is a reusable software module for sorting, data transformation, and high-speed database loading and also unloading.

It is a flexible and extensible tool which adapts at the runtime to the old formats of the given data entered.

It allows creation and the incorporation of new components obtained from any program that gives permission for the integration and reuse of external legacy codes and the search engines.

### **5. Data Profiler:**

Data Profiler is an analytical software that can specify data variety, scope, distribution and variance. It runs in a graphic environment on top of the Co>Op.

I've never used abinitio data profiler, can someone throw some light on how it will help in data quality, as I understand data profiler provides statistical information. Internally it uses abinitio graph. So why we use this at all. Can't we get all those things by using various components.

Ab initio Data Profiler is an analytical application that helps us to specify the given data range, scope, distribution, variance, and the quality. It also runs in a graphical environment on the top of the Co>Operating System.

### **6. Conduct>It:**

It is a powerful GUI-based parallel processing tool for ETL data management and analysis.

Conduct>It's a high-volume knowledge processing systems developing tool. It allows combining graphs from GDE with customized scripts and programs from other vendors.

Ab Initio provides both Graphical and command line interface to Conduct>It.

The Conduct>It® product provides sophisticated job automation and monitoring for Ab Initio applications and other programs, linking them together into a single execution environment. The Conduct>It product can launch applications that run in parallel across distributed sets of Unix, Linux, Windows, and z/OS servers, and containers. It also provides facilities for job alerting, job recovery, and resource management.

Ab initio Conduct>It is a high-volume data processing systems developing tool for developing the data processing in a big manner. It enables us to combine graphs from Graphical Development Environment (GDE) with custom scripts and programs from other vendors. It helps for monitoring and providing the operational statistics for the execution of graphs. It also provides both the graphical and command line interface to the Conduct>It.

**Q. Explain the relation between EME, GDE, and Co-operating system.**

**Enterprise Meta>Environment (EME)**

Is nothing but a repository for storing and managing metadata.

Enterprise Meta-Environment (EME): It's an environment for storage and also metadata management (Both business and technical metadata).

The metadata is accessed from the graphical development environment and also the web browser or the cooperating command line. It is an ab initio repository for any placeholders.

Eme is said as enterprise metadataenv, gde as graphical development env and the co-operating system can be said as abinitio server relation b/w this co-op, eme and gde are as follows operating system is the abinitio server.

**EME:**

- EME stands for Enterprise Metadata Environment
- It is a repository to AbInitio. It holds transformations, database configuration files, metadata and target information

**GDE:**

- GDE – Graphical Development Environment
- It is an end user environment. Graphs are developed in this environment
- It provides GUI for editing and executing AbInitio programs

**Co-operative System:**

- Co-operative system is the server of AbInitio.
- It is installed on a specific OS platform known as Native OS.
- All generated graphs in GDE are later deployed and executed in co-operative system.

**Q. what is the role of Co-operating system in Abinitio?**

- Manage and run Abinitio graph and control the ETL processes
- Provide Ab initio extensions to the operating system
- ETL processes monitoring and debugging
- Meta-data management and interaction with the EME

Co>Operating System operates on the top of the operating system and works as a base for all Ab Initio processes.

It can run on operating systems like Windows, Linux, Solaris, AIX, HP-UX, and z/OS.

It provides Ab Initio extensions through which ETL processes can be controlled. It manages metadata by interacting with EME, manages, and runs Ab Initio graphs.

### **Q. Explain what is SANDBOX?**

A SANDBOX is referred for the collection of graphs and related files that are saved in a single directory tree and behaves as a group for the purposes of navigation, version control, and migration.

A sandbox is like a directory that contains a collection of Ab Initio graphs and related files. It is local to any user and is a replica of the Project in the EME. It will be helpful for version control, migration, and navigation.

A SANDBOX is referred for the compilation of graphs and connected files that are saved in a particular directory tree and behaves as a set for version control, navigation, and relocation.

### **BASIC TERMS of In Ab Initio**

#### **Q. What is Dataset?**

In Simple terms dataset is a file. A file can be main frame file or any fixed or delimited files. There are various **types of datasets**

- FIXED EBCDIC
- FIXED ASCII
- DELIMITED ASCII
- SAS dataset etc...

You can also think of dataset as a table in database world.

#### **Q. What is Component?**

Component is Ab Initio Program

There are various components like SELECT, FILTER, SORT, JOIN, MERGE, DEDUP, ROLLUP, SCAN, USER DEFINED etc....

**Q. What is Port?**

A port is a connection point for the input or output to a component.

**Q. What is Flow?**

A flow carries a stream of data between components in a graph. Flows connect components via ports.

Ab Initio supplies four kinds of flows with different patterns:

straight, fan-in, fan-out, and all-to-all. We will discuss various kinds of flows as we go through this training.

**Q. What is Graph?**

A graph is a diagram that defines the various processing stages of a task and the streams of data as they move from one stage to another. Visually, stages are represented by components and streams are represented by flows. The collection of components and flows comprise an Ab Initio graph.

**Q. What is Field?**

A field is Equivalent to column of a table in Database world. Field is also called variable

**Q. What is Key?**

Key is used many places in Ab Initio Development, We use key to sort the data, Join two files, Rollup data etc...

**Q. List the commonly used components in an Ab Initio graph?**

The commonly used components in an Ab Initio graph are,

- Input file/output file
- Lookup file
- Input table/output table

- Join
- Sort
- Partition
- Partition by key
- Gather
- Reformat
- Concatenate
- Rollup
- Normalize
- Scan
- Dedup
- Filter
- Replicate
- Merge

**Q. What are the components or functions available in ab initio?**

The main components in ab initio are here below,

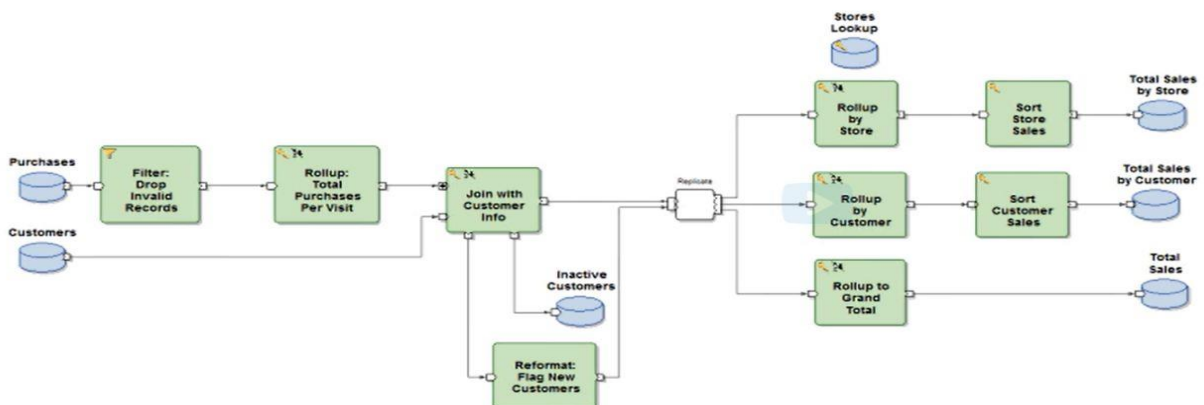
Component	Purpose
Dedup	To remove duplicates
Join	To join multiple input dataset based on a common key value.
Sort	<p>This component reorders the data. It takes the collation order and dumps data to memory.</p> <p>Sort the elements ascending order or descending order</p> <p>Ex:- Score(Ace)</p>
Filter	<p>Any conditional related removal of data.</p> <p>Ex:- mark&gt;75</p>

Replicate	This is component is mainly for parallelism as an additional copy of data is useful while any other nodes go unavailable.
merge	This component is to combine multiple input data.

## Q. Basic Components

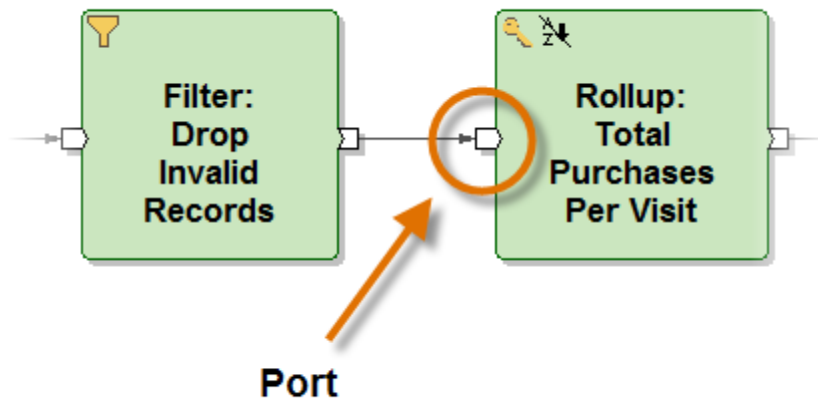
An example graphs

- This graph reads data from three files: one containing information about items purchased, another containing information about customers, and a third with information about stores.
- It produces three sales reports and a file of inactive customers.
- It uses several different kinds of processing components: Filter by Expression, Rollup, Join, Reformat, Replicate, and Sort.



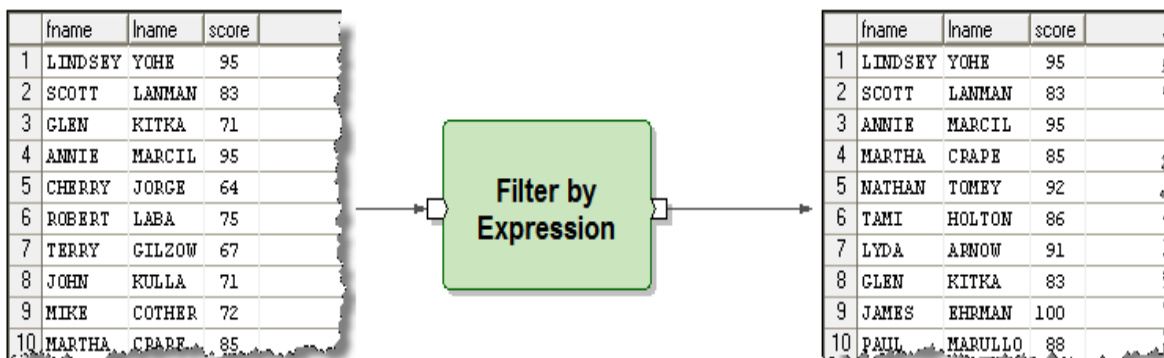
## port

The place where a flow connects to a component is called a port.



## Filter by Expression

The Filter by Expression component reads records from its input port and evaluates a selection expression for each one. If the result is true, it writes that record to its output port.

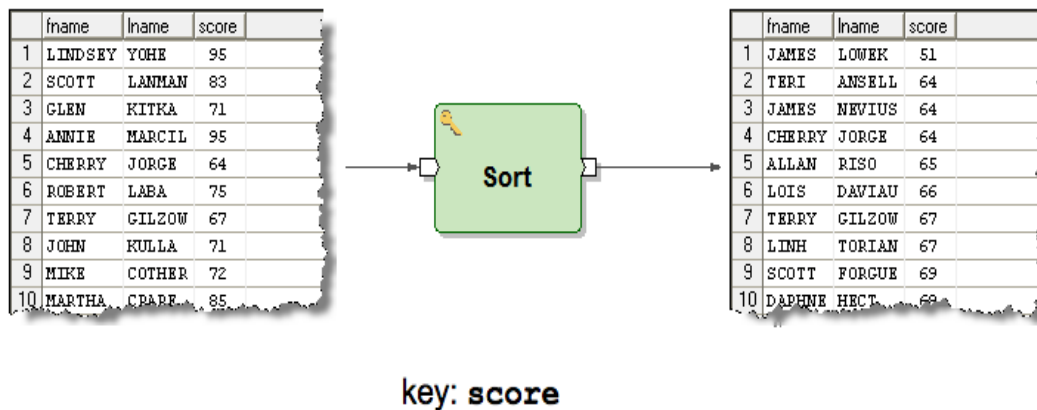


`select_expr: score > 75`



## Sort

The Sort component reads records from its input port, sorts them by one or more key fields, then writes the records in sorted order to its output port.



### Q. Explain what is Sort Component in Abinitio?

The Sort Component in Abinitio re-orders the data. It comprises of two parameters “Key” and “Max-core”.

#### Key:

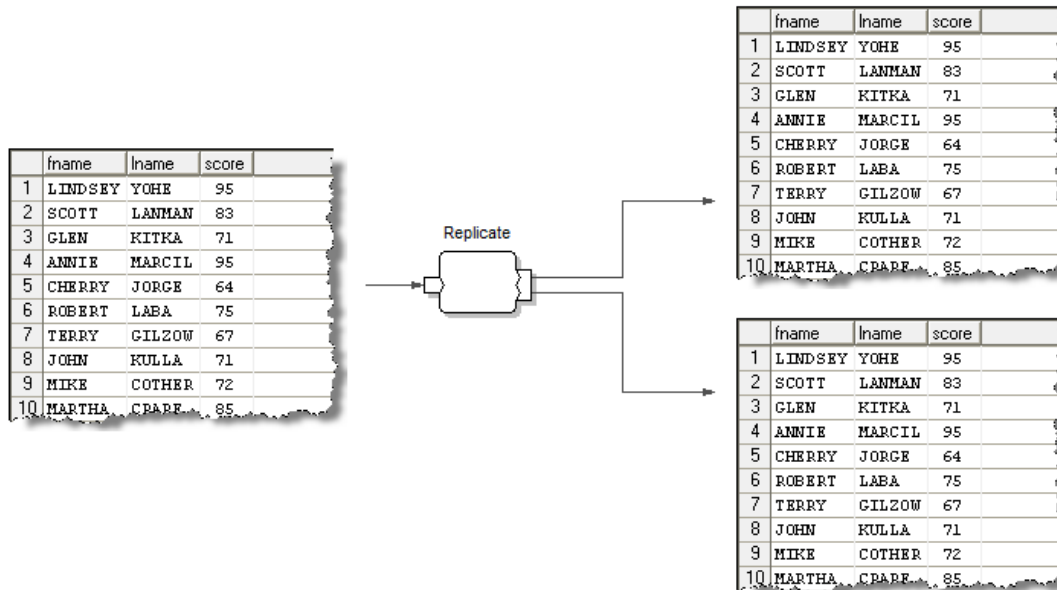
- It is one of the parameters for sort component which determines the collation order
- It represents the collation order.

#### Max-core:

- This parameter controls how often the sort component dumps data from memory to disk.
- It defines how often the sort component should dump data from memory to disk.

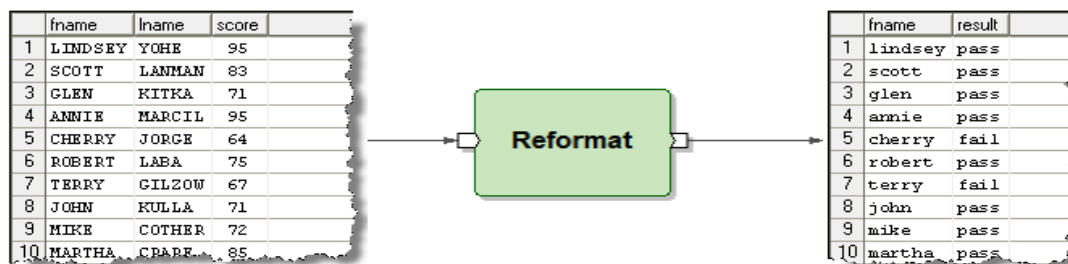
## Replicate

The Replicate component simply copies every record arriving on its input port to every flow connected to its output port.



## Reformat

The Reformat component reads records from its input port and, for each, applies a set of rules to create a new record, which it writes to its output port.

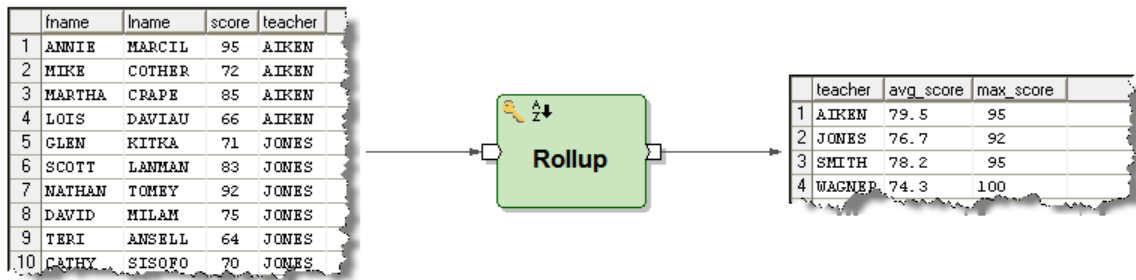


transform:

```
string_lowercase(fname)           → fname
if (score > 70) "pass" else "fail" → result
```

## Rollup

The Rollup component aggregates collections of records grouped by a key. It applies a set of rules to produce an output record for each group.



transform:

```
teacher      → teacher
avg(score)   → avg_score
max(score)   → max_score
```

**Q. What is Rollup Component?**

**Q. What do you understand by roll-up component?**

Roll-up component enables the users to group the records on certain field values. It is a multiple stage function and consists initialize 2 and Rollup 3.

A rollup component is used to group the records based on certain field values. It is a multi-stage transform function which contains functions like initialize, rollup and finalize.

The roll-up component facilitates users to collect or group the records on certain field values. It is called for each of the records in the group and consists of initializing 2 and Rollup 3.

**Q. Have you used the rollup component? Describe how?**

If the user wants to group, the records on particular field values then rollup is the best way to do that. Rollup is a multi-stage transform function and it contains the following mandatory functions.

1. initialize
2. rollup

### 3. finalize

Also, need to declare one temporary variable if you want to get counts of a particular group.

For each of the group, first, it does call the initialize function once, followed by rollup function calls for each of the records in the group and finally calls the finalize function once at the end of last rollup call.

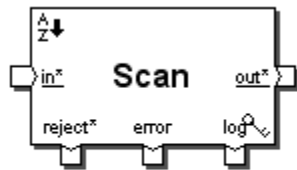
\*\* Rollup Component has been used for grouping the records which are based on Field Values. It also is a Multi-Stage Transforming function that contains functions such as:

Initialize

Rollup

Finalize

## Scan Component



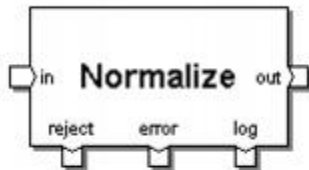
### Abinitio Coach

For every input record, Scan generates an output record that consists of a running cumulative summary for the group to which the input record belongs, up to and including the current record.

**For example**, the output records might include successive year-to-date totals for groups of records.

## Normalize Component

Normalize component is also one of the multistage components in ab initio.



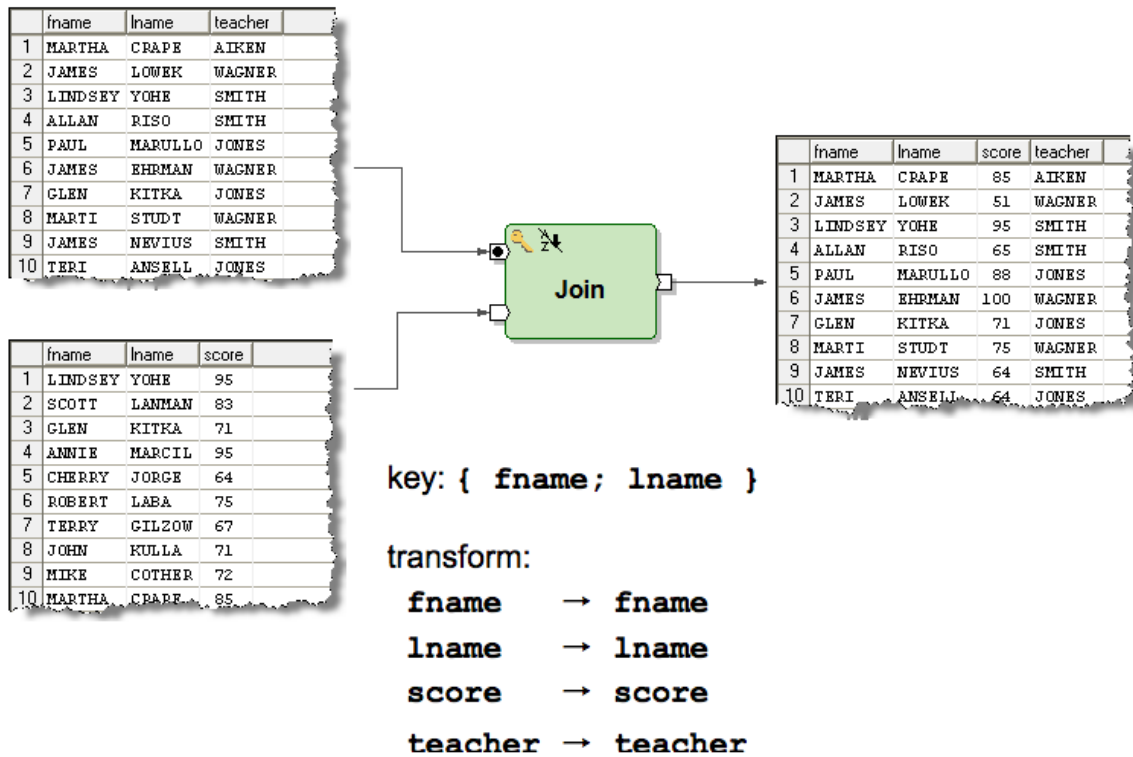
### Abinitio Coach

Normalize generates multiple output records from each of its input records. You can directly specify the number of output records for each input record, or you can make the number of output records dependent on a calculation.

In contrast, to consolidate groups of related records into a single record with a vector field for each group — the inverse of NORMALIZE — you would use the accumulation function of the ROLLUP component.

## Join

The Join component combines the data arriving on its input ports. Each pair of input records with matching keys is combined using rules to produce an output record.



**Q. In how many segments the Abinitio EME can be segregated?**

**Q. Explain how Abinitio EME is segregated?**

- Data Integration Portion
- User Interface ( Access to the meta-data information)

The Ab Initio EME is logically segregated into data integration portion and user interface to access metadata information

**Q. Mention how can you connect EME to Abinitio Server?**

To connect with Ab initio Server, there are several ways like

1. Login to EME web interface-  
`http://serverhost:[serverport]/abinitio` to connect EME to Abinitio Server.
2. Set AB\_AIR\_ROOT
3. We can connect to the EME data store through GDE.
4. We can also use air-command to connect EME to Abinitio Server.

**Q. How to connect EME to Ab Initio Server?**

EME can be connected to the Ab Initio server in several ways. The following are the ways to connect to EME.

1. Set AB\_AIR\_ROOT
2. Air commands
3. EME web interface at [http://serverhost:\[serverport\]/abinitio](http://serverhost:[serverport]/abinitio)
4. Connect to EME data store through GDE

**For connecting Eme with Ab Initio, there are many ways such as:**

1. We can login to EME Web Interfaces.
2. We can connect to EME Data Store.
3. By going through Air Command.
4. By setting AB\_AIR\_ROOT.

**Q. What are the types of output formats that we can get after processing data?**

- Charts
- Tables
- Vectors
- Plain Text files
- Maps
- Raw files
- Image files

**Q. Explain what does dependency analysis mean in Abinitio?**

Dependency Analysis is used for retrieving the maximum from the existing data and by using next\_in\_sequence/reformat we can also generate the sequence. The data comes from dependency analysis, the applications can depend on and produce this data, etc.

In Ab initio, dependency analysis is a process through which the EME examines a project entirely and traces how data is transferred and transformed- from component-to-component, field-by-field, within and between graphs.

The dependency analysis is a process through which the EME analyzes the project for the dependencies within and between the graphs. It examines the entire project and tracks how data is being transformed and transferred from component to component and field by field. The steps involved in dependency analysis are translation and analysis.

Dependency analysis is a procedure through which the EME examines a project completely and traces how data is transferred and distorted from field-by-field, component-to-component, within and among graphs.

Dependency analysis will answer the questions regarding data lineage. That is where does the data come from, what applications produce and depend on this data etc.

We can retrieve the maximum (surrogate key) from the existing data, the by using the scan or next\_in\_sequence/reformat we can generate a further sequence for new records. ([E Learning pottal](#)).

In Abinitio, dependency analysis is a process through which the EME examines a project entirely and traces how data is transferred and transformed- from component-to-component, field-by-field, within and between graphs.



**Q. What are the different types of parallelism used in Abinitio?**

**Parallelism is the concurrent execution of multiple components, and it's a good way to increase performance.**

Different types of parallelism used in Abinitio includes

- **Component parallelism:**

The component parallelism is used by a graph with multiple processes executing simultaneously on separate data.

A graph with multiple processes executing simultaneously on separate data uses parallelism

**It is used by the graph that has multiple processes executing simultaneously on separate data.**

An application that has multiple components running on the system simultaneously. But the data are separate. This is achieved through component level parallel processing.

- **Data parallelism:**

The data parallelism is used by a graph that works with data divided into segments and operates on each segment respectively.

A graph that works with data divided into segments and operates on each segments respectively, uses data parallelism.

It is used by the graph that works with data divided into segments, which operates on each segment, respectively.

Data is split into segments and runs the operations simultaneously. This kind of process is achieved using data parallelism.

- **Pipeline parallelism:**

A graph that deals with multiple components executing simultaneously on the same data uses pipeline parallelism. Each component in the pipeline reads continuously from the upstream components, processes data and writes to downstream components. Both components can operate in parallel.

It is used by the graph that deals with multiple components executing simultaneously on the same data. An application with multiple components but running on the same dataset. This uses pipeline parallelism.

### **Q. What Parallelisms Does Abinitio Support?**

AbInitio supports 3 parallelisms. They are

- **Data Parallelism :** Same data is parallelly worked in a single application
- **Component Parallelism :** Different data is worked parallelly in a single application
- **Pipeline Parallelism :** Data is passed from one component to another component. Data is worked on both of the components.

**Q. What is the different way to achieve the partitions?**

There are multiple ways to do the partitions.

Partitions	Description
Expression	Data split according to the data manipulation language.
Key	Grouping the data by specific keys
Load balance	Dynamic load balancing
Percentage	Segregate the data where the output size is on the fractions of 100
Range	Split the data evenly based on a key and a range among the nodes
Round robin	Distributing the data evenly in blocksize across the output partitions.

1. **Partition by Round-Robin:** Distributing data evenly, in block size chunks, across the output partitions
2. **Partition by Range:** You can divide data evenly among nodes, based on a set of partitioning ranges and key
3. **Partition by Percentage:** Distribution data, so the output is proportional to fractions of 100
4. **Partition by Load balance:** Dynamic load balancing
5. **Partition by Expression:** Data dividing according to a DML expression
6. **Partition by Key:** Data grouping by a key

**Q. What are the types of partition components in Ab Initio?**

**Q. Mention what is a partition and what are the different types of partition components in Abinitio?**

In Abinitio, partition is the process of dividing data sets into multiple sets for further processing. Different types of partition component includes

**1. Partition by Round-Robin:**

- The Round-Robin Partition is used for distributing data evenly, in block size chunks, across the output partitions.
- Distributing data evenly, in block size chunks, across the output partitions.
- It distributes the data evenly in block size chunks.

**2. Partition by Range:**

- The Partition by Range facilitates users to divide data evenly among nodes, according to the set of partitioning ranges and keys.
- You can divide data evenly among nodes, based on a set of partitioning ranges and key.
- Based on a set of partitioning ranges and key, it divides data evenly among nodes.

**3. Partition by Percentage:**

- The Partition by Percentage is used to distribute data in a way that the output is proportional to fractions of 100.
- Distribution data, so the output is proportional to fractions of 100.
- It distributes data in such a way that the output is proportional to fractions of 100.

**4. Partition by Load balance:**

- The Partition by Load balance is used for dynamic load balancing.
- Dynamic load balancing.
- It distributes data based on dynamic load balancing.

**5. Partition by Expression:**

- The Partition by Expression is used to divide data according to a DML expression.
- Data dividing according to a DML expression.
- It divides data based on a DML expression.

#### 6. Partition by Key:

- The Partition by Key is used to group data by a key.
- Data grouping by a key.
- It partitions data based on a key.

#### Q. What is the difference between partitioning with key and round-robin?

##### Partition by key:

In this, we have to specify the key based on which the partition will occur. It results in well-balanced data due to the key-based partitions. It is useful for key- dependent, parallelism.

##### Partition by round-robin:

In this, distributing data evenly in block size chunks, the records are partitioned in a sequential way across the output partition. It is not key-based, and results are well-balanced data, especially with a block size of 1. It is useful for record independent parallelism.

#### Q. Explain what is de-partition in Abinitio?

#### Q. What do you understand by de-partition in Abinitio?

De-partition is used to read data from multiple flows or operations and re-join data records from different flows. There are Several de-partition components are available in Abinitio, such as Gather, Merge, Interleave, Concatenation, etc.

De-partition helps in allowing us to read from multiple flow and operations, is used for re-joining data records from different flows. De-partition consists of components that include:



**Gather**  
**Merge**  
**Interleave**  
**Concatenation**

### Q. List out the file extensions used in Abinitio?

The file extensions used in Abinitio are

- .mp: This file extension is used to store Abinitio graph or graph components.
- .mpc: This file extension is used to specify a custom component or program.
- .mdc: This file extension is used to specify data-set or custom data-set components.
- .dml: This file extension is used to specify data manipulation language file or record type definition.
- .xfr: This file extension is used to specify transform function files.
- .dat: This file extension is used to specify data files (multifile or serial file).
- .mp: It stores Ab initio graph or **graph component**
- .mpc: **Custom component or program**
- .mdc: Dataset or custom **data-set component**  
dataset template files or custom dataset components
- .dml: **Data manipulation language** file or record type definition  
record format files
- .xfr: Transform **function file**
- .dat: **Data file** (multifile or serial file)
- .dbc - **database table** files
- .ksh - **shell scripting file**

### Q. List out some of the air commands used in Abintio?

Air command used in Abinitio includes

1. **air object ls<EME path for the object-/Projects/edf/..>** : It is used to see the listings of objects in a directory inside the project
2. **air object rm<EME path for the object-/Projects/edf/..>** : It is used to remove an object from the repository
3. **air object versions-verbose<EME path for the object-/Projects/edf/..>** : It gives the version history of the object.

Other air command for Abinitio include air object cat, air object modify, air lock show user, etc.

### **Q. What are the air commands used in Ab Initio?**

- Air object ls
- Air object rm
- Air project modify
- Air object versions -verbose
- Air lock show -user
- Air sandbox status

### **Q. What is a local lookup?**

- Local lookup file has records which can be placed in main memory
- They use transform function for retrieving records much faster than retrieving from the disk.

### **Q. Mention what the difference between “Look-up” file and “Look is up” in Abinitio?**

#### **Lookup file**

Lookup file defines one or more serial file (Flat Files); it is a physical file where the data for the Look-up is stored.

A lookup file represents one or more serial files, also known as Flat files. It is a physical file where lookup data is stored, which is small enough to be held in memory.

#### **Look-up**

While Look-up is the component of abinitio graph, where we can save data and retrieve it by using a key parameter.

Lookup is a component of the Ab Initio graph where data resides along with a key parameter. The data can be retrieved using this key parameter.

### **Q. What is a local lookup?**

The local lookup function will be used before the lookup function call when the lookup file is a multifile and partitioned/sorted on a particular key.

It will be local to a partition, depending on the key. The data records in the lookup file can be loaded into memory. This way, the transform function retrieves records faster than retrieving from disk.

### Q. What Is A Local Lookup?

- Local lookup file has records which can be placed in main memory
- They use transform function for retrieving records much faster than retrieving from the disk.

### Q. Define a local lookup?

Local lookup file has documentations which can be located in major memory, and they are also used to change function for retrieving records much earlier than retrieving from the disk.

### Q. What do you understand by the "lookup" file in Abinitio?

In Abinitio, the lookup file is used to define one or more serial files (also known as flat files). It is a physical file that stores the data for the Lookup. It is a two-dimensional table of data that has been stored in a disk file. It stores the name and display format for each column of data depending on the file format.

### Q. What is a ramp limit?

- A limit is an integer parameter which represents a number of reject events
- Ramp parameter contain a real number representing a rate of reject events of certain processed records
- The formula is – No. of bad records allowed = limit + no. of records x ramp
- A ramp is a percentage value from 0 to 1.
- These two provides the threshold value of bad records.

### Q. What is the relation between limit and ramp?

Limit and ramp are used to set the reject tolerance of a graph.

Limit is the number of rejects and ramp is rate of rejection.

The formula for rejects tolerance is, **limit + (ramp\*no\_of\_records\_processed)**



### Q. What is a ramp limit?

- A limit is an integer parameter which represents several reject events
- Ramp parameter contains a real number representing a rate of reject events of certain processed records.
- The formula is – No. of bad records allowed = limit + no. of records x ramp.
- A ramp is a percentage value from 0 to 1.
- These two provides the threshold value of bad records.

### Q. Define ramp limit in AB Initio?

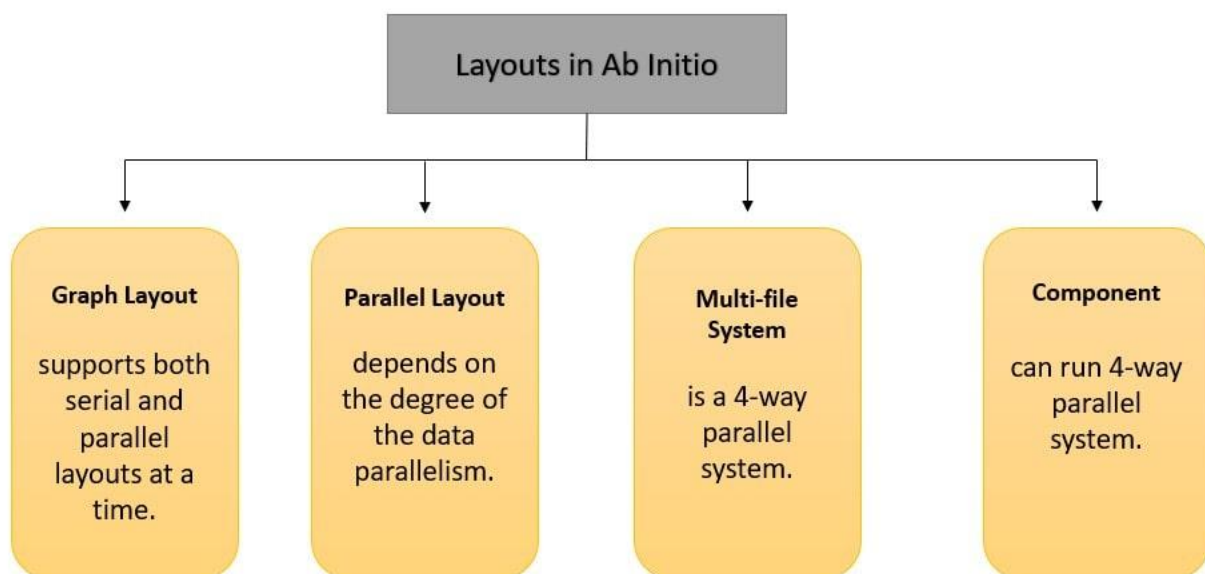
Generally, the ramp is referred to the percentage value ranging from 0 to 1. To represent the number of reject events, the limit parameter possesses an integer called a ramp limit.

### Q. What do you mean by ramp limit?

An edge is an integer parameter which represents a numeral of reject events. Ramp parameter encloses a real number on behalf of a rate of decline events of certain processed records; it has a percentage value from 0 to 1.

### Q. What are the kinds of layouts that Ab Initio supports?

A layout defines which component should run where. Ab Initio has two kinds of layouts.



**Serial layout** - the level of parallelism is 1.

**Parallel layout** - the level of parallelism depends on the data partition.

**Q. What kind of layouts does Ab Initio support?**

- Supports serial and parallel layouts.
- A graph layout supports both serial and parallel layouts at a time.
- A multi-file system is a 4-way parallel system.
- A component in a graph system can run 4-way parallel system.

**Q. Mention the kind of layouts does Abinitio supports?**

There are successive and parallel layouts supported by Abinitio. A graph can have both at the same time. The parallel one depends on the amount of data parallelism. If the multi-file system is 4-way parallel, then a part in a graph can run four-way parallel if the layout is distinct such as it's same as the degree of parallelism.

**Q. How to improve the performance of a graph?**

We can improve the performance of a graph through the following methods.

- Use lookup instead of join and merge components.
- When we have to join two files and don't want duplicates, use a union function instead of a duplicate remover.
- Minimize the use of sort components.
- Reduce the use of regular expression functions in the transfer functions.
- Don't use broadcast as partitioner for large datasets.
- Use only the fields that are required in sort, reformat, and join components.

**Q. How do you improve the performance of a graph?**

- Reduce the usage of multiple components on certain phases.
- Use a refined and well-defined value of max core values for sort and join components.
- Minimize the use of regular expression functions like `re_index` in the transfer functions

- Minimize sorted join component and, if possible, replace them with in-memory join/hash join.
- Use only required fields in the sort, reformat, join components.
- Using Phase or the flow buffering during the cases of merge or sorted joins.
- Use hash join if the two sets of input are small; else better to choose the sorted join for the huge input size.
- For large dataset better not use broadcast as partitioned
- Reduce the number of sort components while processing.
- Avoid repartitioning of data unnecessarily.

**Q. Explain how you can run a graph infinitely in Ab initio?**

To execute graph infinitely, the graph end script should call the .ksh file of the graph. Therefore, if the graph name is abc.mp then in the end script of the graph it should call to abc.ksh. This will run the graph for infinitely.

Calling the .ksh file of the graph at the end script runs the graph infinitely. If the graph name is xyz.mp, then the end script of the graph should make a call to xyz.ksh.

**Q. Is it possible to run a graph infinitely in Ab Initio? If yes, how?**

Yes, it is possible to run a graph infinitely in Ab Initio. To do so, the graph end script should call the .ksh file of the graph. After that, if the graph name is xyz.mp then in the end script of the graph, it should call to xyz.ksh. By following the above steps, we can run the graph for infinitely.

**Q. What do you mean by the overflow errors?**

The overflow errors are the errors that occur when the computer cannot process the bulk data. While processing data, overflow errors occur if the bulky calculations exceed the range of memory provided to them.

Overflow errors are the errors raised when dealing with processing bulky sets of data. While processing data, bulky calculations might not fit the memory allocated for them. And when a character of more than 8-bits is stored, an overflow error is raised.

While processing data, bulky calculations are often there and it is not always necessary that they fit the memory allocated for them. In case a character of more than 8-bits is stored there, these errors result simply.

**Q. Mention what dedup-component and replicate component does?**

**Dedup component:**

- It is used to remove duplicate records.
- It is used to remove duplicate records from the flow based on a specified key.

**Replicate component:**

- It combines the data records from the inputs into one flow and writes a copy of that flow to each of its output ports.
- It is used to combine input records from multiple sources into one flow and write a copy of that flow to output ports.

**Q. What is the usage of dedup component and replicate component in Abinitio?**

In Abinitio, the dedup component is used to eliminate duplicate records. On the other hand, the replicate component combines the data records from the inputs into one run and writes a copy of that run to each of its output ports.

**Q. Mention what information does a .dbc file extension provides to connect to the database?**

- The .dbc extension provides the GDE with the information to connect with the database are
- Name and version number of the data-base to which you want to connect
- Name of the computer on which the data-base instance or server to which you want to connect runs, or on which the database remote access software is installed
- Name of the server, database instance or provider to which you want to link

**Q. What information does a .dbc file extension provide to connect to the database?**

- The .dbc file provides the below information to connect to a database.
- Name and version number of the database
- Name of the system on which the database server or instance is running.
- Name of the server or database instance

### Q. What are `is_valid` and `is_define` used for?

**`is_valid`** - It is used to test if a value is valid or not. If the expression is a valid data item, the value will be 1. If the expression is not a valid data item, the value will be 0.

**`is_define`** - It is used to test if an expression is not NULL. If the expression is non NULL, the value will be 1. The value will be 0 otherwise.

### Q. Mention what is the syntax for `m_dump` in Abinitio?

The syntax for `m_dump` in Abinitio is used to view the data in multifile from unix prompt. The command for `m_dump` includes

**`m_dump a.dml a.dat`**: This command will print the data as it manifested from GDE when we view data in formatted text

**`m_dump a.dml a.dat>b.dat`**: The output is re-directed in `b.dat` and will act as a serial file. `b.dat` that can be referred when it is required.

### Q. How to add default rules in the transformer?

Go to component properties, navigate to the parameter tab page, and double click on the transform parameter.

The transform editor page will open. Click on the edit menu and select the 'Add Default Rules' option from the drop-down. You can choose from Match names and Wildcard options.

- Double click on the transform parameter in the parameter tab page in component properties
- Click on Edit menu in Transform editor
- Select Add Default Rules from the dropdown list box.
- It shows Match Names and Wildcard options. Select either of them.

### Q. What is a surrogate key?

The system generated unique sequential number is called a surrogate key. It acts as a primary key.

**Q. How will you handle it if DML is changing dynamically?**

We have a lot of ways through which we can handle dynamically changing DML. Some of the methods are,

- Use a conditional DM
- Call the vector functionality while calling the DMLs
- Use the MULTI REFORMAT component

**Q. What is Data Encoding?**

Data Encoding is used for keeping many cases confidential, it also is used for making sure if the information remains in the form by which no one else other than the receiver and the sender will understand.

**Q. How data is processed and what are the fundamentals of this approach?**

There are certain activities which require the collection of the data and the best thing is processing largely depends on the same in many cases. The fact is data needs to be stored and analyzed before it is processed. This task depends on some major factors are they are

1. Collection of Data
2. Presentation
3. Final Outcomes
4. Analysis
5. Sorting

These are also regarded as the fundamentals that can be trusted to keep up the pace in this matter.

**Q. What are the factors on which storage of data depends?**

It depends on the sorting and filtering. In addition to this, it largely depends on the software one uses.

**Q. What do you mean by data sorting?**

It is not always necessary that data remains in a well-defined sequence. It is always a random collection of objects. Sorting is nothing but arranging the data items in desired sets or sequence.

**Q. Explain data sorting?**

It is not always essential that data remains in a well-defined series. It is always a chance compilation of objects. Sorting is nil but arranging the data items in preferred sets or progression.

**Q. When running a stored procedure definition script how would you guarantee the definition could be rolled back in the event of problems?**

There are quite a few factors that determine the approach such as what type of version control is used, what is the size of the change, what is the impact of the change, is it a new procedure or replacing an existing and so on.

If it is new, then just drop the wrong one.

if it is a replacement then how big is the change and what will be the possible impact, depending upon you can have the entire database backed up or just create a script for your original procedure before messing it up or you just do ed and change the file back to original and reapply. you may rename the old procedure as old and then work on new and so on.

few issues to keep in mind are synonyms, dependencies, grants, any job calling the procedure at the time of change and so on. In nutshell, the scenario can be varied and the solution also can be varied.

**Q. What is AB\_LOCAL expression where do you use it in ab-initio?**

ablocal\_expr is a parameter of the table component of Ab Initio.LOCAL() is replaced by the contents of ablocal\_expr. Which we can make use in parallel unloads. There are two forms of AB\_LOCAL() construct, one with no arguments and one with a single argument as a table name(driving table).

The use of AB\_LOCAL() construct is in Some complex SQL statements contain grammar that is not recognized by the Ab Initio parser when unloading in parallel. You can use the LOCAL() construct in this case to prevent the Input Table component from parsing the SQL (it will get passed through to the database). It also specifies which table to use for the parallel clause.

**Q. How to get DML using Utilities in UNIX?**

If your source is a COBOL copybook, then we have a command in Unix which generates the required in Ab Initio.

**Q. State the first\_defined function with an example?**

This function is similar to the function NVL() in Oracle database

It performs the first values which are not null among other values available in the function and assigns to the variable

**Example:** A set of variables, say v1,v2,v3,v4,v5,v6 are assigned with NULL.

Another variable num is assigned with value 340 (num=340)

num = first\_defined(NULL, v1,v2,v3,v4,v5,v6,NUM)

The result of num is 340.

**Q. How does the force\_error function work?**

If any mentioned conditions are not met, the force\_error it forces an error. It will be useful when you want to stop the execution of a graph if it doesn't meet the set condition.

It will send the records to the reject port and error message to the error port.

**Q. What is a multfile system?**

Multifile is a set of directories on different nodes in a cluster. They possess an identical directory structure. The multifile system leads to better performance as it is parallel processing where the data resides on multiple disks.

It is created with the control partition on one node and data partitions on the other nodes to distribute the processing in order to improve the performance.

**Q. What is data encoding in Abinitio?**

Data needs to be kept classified in many cases, and it can be complete through this approach. It ensures information scraps in a form which no one else than the sender and the receiver can appreciate.

Data needs to be kept confidential in many cases and it can be done through this approach. It



simply makes sure of information remains in a form which no one else than the sender and the receiver can understand.

In Abinitio, data encoding is an approach that is used to keep data confidential. In this approach, we ensure that the information remains in a form that cannot be understood by someone else other than the sender and the receiver.

**Q. Define the evaluation of parameters order?**

**Q. Describe the Evaluation of Parameters order?**

Following is the order of evaluation:

- The host setup script will be executed first
- All Common parameters, that is, included, are evaluated
- All Sandbox parameters are evaluated
- The project script – project-start.ksh is executed
- All form parameters are evaluated
- Graph parameters are evaluated
- The Start Script of the graph is executed

**Q. Define local and formal parameter?**

Both are graph level parameters, but in local you need to initialize the worth at the time of announcement whereas globe no need to initialize the data it will produce at the time of operation the graph for that parameter.

**Q. Name the operations that support avoiding duplicate record?**

- Performing aggregation
- Utilizing the Rollup part
- Using Dedup class

**Q. What is the use of aggregation when we have rollup as we know rollup component in abinitio is used to summarize a group of data record? Then where we will use aggregation?**

Aggregation and Rollup both can summarize the data but rollup is much more convenient to use. To understand how a particular summarization being rollup is much more explanatory compared to aggregate.

Rollup can do some other functionality like input and output filtering of records. Aggregate and rollup perform the same action, rollup display intermediate result in main memory, Aggregate does not support intermediate result.

**Q. Describe the elements you would review to ensure multiple scheduled batch jobs do not collide with each other?**

Because every job depends upon another job for example if your first job result is successfully then another job **will execute otherwise your job doesn't work.**

**Q. How to create a repository in abinitio for stand-alone system(LOCAL NT)?**

If you are trying to install the Ab -Initio on a stand-alone machine, then it is not necessary to create the repository, While installing It creates automatically for you under abinitio folder ( where you installing the Ab-Initio) If you are still not clear please ask your Question on the same portal.

**Q. Describe the process steps you would perform when defragmenting a data table. Does this table contain mission-critical data?**

1) We can move the table in the same or other tablespace and rebuild all the indexes on the table. alter table move this activity reclaims the defragmented space in the table analyze table table\_name compute statistics to capture the updated statistics.

2) Reorg could be done by taking a dump of the table, truncate the table and import the dump back into the table.

**Q. Why might you create a stored procedure with the with recompile option?**

Recompile is useful when the tables referenced by the stored proc undergoes a lot of modification/deletion/addition of data. Due to the heavy modification activity, the execute plan becomes outdated and hence the stored proc performance goes down.

If we create the stored proc with recompile option, the sql server won't cache a plan for this stored proc and it will be recompiled every time it is run.

**Q. Explain PDL with an example?**

To make a graph behave dynamically, PDL is used

Suppose there is a need to have a dynamic field that is to be added to a predefined DML while executing the graph

Then a graph level parameter can be defined

Utilize this parameter while embedding the DML in the output port.

**For Example:** define a parameter named field with a value "string(" | "(") name;"

Use \${mystring} at the time of embedding the DML in out port.

Use \$substitution as an interpretation option.

**Q. What is BROADCASTING and REPLICATE?**

**Broadcast** – Takes data from multiple inputs, combines it and sends it to all the output ports.

Eg – You have 2 incoming flows (This can be data parallelism or component parallelism) on Broadcast component, one with 10 records & other with 20 records. Then on all the outgoing flows (it can be any number of flows) will have  $10 + 20 = 30$  records.

**Replicate** – It replicates the data for a particular partition and send it out to multiple outputs of the component, but maintains the partition integrity.

**Eg** – Your incoming flow to replicate has a data parallelism level of 2. with one partition having 10 recs & other one having 20 recs.

Now suppose you have 3 output flows from replicate. Then each flow will have 2 data partitions with 10 & 20 records respectively.

**Q. What are the benefits of data processing according to you?**

Well, processing of data derives a very large number of benefits. Users can put separate many factors that matter to them. In addition to this, with the help of this approach, one can easily keep up the pace simply by deriving data into different structures from an unstructured format.

In addition to this, the processing is useful in eliminating various bugs that are often associated with the data and cause problems in a later section.

It is because of no other reason than this, data processing has a wide application in several tasks.

**Q. What exactly do you understand with the term data processing and businesses can trust this approach?**

Processing is a procedure that simply covert the data from a useless form into a useful one without making a lot of efforts. However, the same may vary depending on factors such as the size of data and its format.

A sequence of operations is generally carried out to perform this task and depending on the type of data, this sequence could be automatic or manual. Because in the present scenario, most of the devices that perform this task are PC's automatic approach is more popular than ever before.

Users are free to obtain data in forms such as a table, vectors, images, graphs, charts and so on. This is the best things that business owners can simply enjoy.

**Q. Describe how you would ensure that database object definition (Tables, Indices, Constraints, Triggers, Users, Logins, Connection Options, and Server Options, etc) are consistent and repeatable between multiple database instances (i.e.: a test and production copy of a database)?**

Take an entire database backup and restore it in a different instance.

Take statistics of all valid and invalid objects and match.

Periodically refresh.

**Q. What is local and formal parameter?**

Two are graph level parameters but in local you need to initialize the value at the time of declaration whereas global no need to initialize the data it will print at the time of running the graph for that parameter.

**Q. What Is Max Core Of A Component?**

- MAX CORE is the space consumed by a component that is used for calculations
- Each component has different MAX COREs
- Component performances will be influenced by the MAX CORE's contribution
- The process may slow down / fasten if a wrong MAX CORE is set

**Q. How can we find old recovery files on Ab Initio production boxes?**

We can find the recovery files by using the following command:

```
find . -name \*.rec
```

### **Q. What Is The Function That Transfers A String Into A Decimal?**

Use decimal cast with the size in the transform() function, when the size of the string and decimal is same.

Ex: If the source field is defined as string(8).

- The destination is defined as decimal(8)
- Let us assume the field name is salary.
- The function is out.field :: (decimal(8)) in salary
- If the size of the destination field is lesser than the input then string\_substring() function can be used

Ex : Say the destination field is decimal(5) then use...

- out.field :: (decimal(5))string\_ltrim(string\_substring(in.field,1,5))
- The ' ltrim ' function is used to remove leading and trailing spaces in the string

### **Q. Explain Pdl With An Example?**

To make a graph behave dynamically, PDL is used

- Suppose there is a need to have a dynamic field that is to be added to a predefined DML while executing the graph
- Then a graph level parameter can be defined
- Utilize this parameter while embedding the DML in output port.

For Example : define a parameter named myfield with a value "string(" | "(") name;"

- Use \${mystring} at the time of embedding the dml in out port.
- Use \$substitution as an interpretation option

**Q. State The Working Process Of Decimal\_strip Function.**

- A decimal strip takes the decimal values out of the data.
- It trims any leading zeros
- The result is a valid decimal number

**Ex:**

`decimal_strip("-01840") := "-184"`

`decimal_strip("oxyas97abc") := "97"`

`decimal_strip("+ $78ab=-&^&%cdw") := "78"`

`decimal_strip("Honda") "0"`

**Q. State The First\_defined Function With An Example.**

- This function is similar to the function NVL() in Oracle database
- It performs the first values which are not null among other values available in the function and assigns to the variable

**Example:** A set of variables, say v1,v2,v3,v4,v5,v6 are assigned with NULL.

Another variable num is assigned with value 340 (num=340)

`num = first_defined(NULL, v1,v2,v3,v4,v5,v6,NUM)`

The result of num is 340

**Q. What Are The Operations That Support Avoiding Duplicate Record?**

Duplicate records can be avoided by using the following:

- Using Dedup sort
- Performing aggregation
- Utilizing the Rollup component

**Q. What Is A Deadlock And How It Occurs?**

- A graphical / program hand is known as deadlock.
- The progression of a program would be stopped when a dead lock occurs.
- Data flow pattern likely causes a deadlock
- If a graph flows diverge and converge in a single phase, it is potential for a deadlock
- A component might wait for the records to arrive on one flow during the flow converge, even though the unread data accumulates on others.
- In GDE version 1.8, the occurrence of a dead lock is very rare.

**Q. What is the component that can be used to lower a file in size?**

The components that can be used are Deflate and Compress.

**Q. Can a graph be infinitely run? If yes, how?**

Yes it can, by calling the .ksh file in the end of the script.

**Q. What meaning has lock in Ab Initio?**

A graph must be locked in order to give permission only to the developer for editing it or the objects that belong to the graph. When other users try to lock the graph they will be warned on the monitor that another user has already made a lock. This is a protection mechanism for developers.

**Q. What possible errors we can receive from a graph execution?**

The compilation errors in Ab Initio can appear when for example the output DML is not as the input DML.

**Q. What role has the XFR function?**

The purpose of this function is usually to store mappings and this is very useful because the rewriting of code takes time and the XFR spares us of this effort.

**Q. How much memory do we need for a graph?**

Some calculations lead to approximately eight MB plus the MAX\_CORE and the phase file size.



**Q. How the term Standard Environment can be explained?**

The Standard Environment is the type of environment with more than one project that is private and another public one.

**Q. State the difference between check-point and phase?**

The difference between them is:

**CHECKPOINT:**

- The break of the procedure will be sustained following the checkpoint
- Data from the checkpoint is gathered and carry on executing after correction.
- When a graph fails in the centre of the procedure, a recovery point is shaped known as a Checkpoint

**PHASE:**

- All the phases will sprint one by one
- The middle file will be deleted
- If a graph is shaped with phases, each phase is assigned to some division of memory one after another.

**Q. What is the difference between check point and phase in Ab Initio?**

A list of differences between check point and phase in Ab Initio:

Check point	Phase
A check point is a recovery point that is created when a graph fails in the middle of the process.	A graph consists of phases. If a graph is created with phases, each phase is assigned to some part of the memory.
The rest of the process will be continued after the check point.	All the phases run one by one.
Data from the check point is fetched and continues to execute after correction.	In phase, the intermediate file will be deleted.

**Q. Describe the Grant/Revoke DDL facility and how it is implemented?**

GRANT means permissions for example GRANT CREATE TABLE, CREATE VIEW AND MANY MORE.

REVOKE means cancel the grant (permissions). So, Grant or Revoke both commands depend upon D.B.A.

**Q. Explain the difference between the truncate and delete commands?**

**Truncate:**

It is a DDL command, used to delete tables or clusters. Since it is a DDL command hence it is auto commit and Rollback can't be performed. It is faster than delete.

**Delete:**

It is DML command, generally used to delete a record, clusters or tables. Rollback command can be performed, to retrieve the earlier deleted things. To make deleted things permanently, "commit" command should be used.

**Q. What are the primary keys and foreign keys?**

In RDBMS the relationship between the two tables is represented as Primary key and foreign key relationship. Whereas the primary key table is the parent table and the foreign key table is the child table. The criteria for both the tables are there should be a matching column. company

**Q. When using multiple DML statements to perform a single unit of work, is it preferable to use implicit or explicit transactions, and why?**

Because implicit is using for internal processing and explicit is using for using open data required.

**Q. What is an outer join?**

An outer join is used when one wants to select all the records from a port – whether it has satisfied the join criteria or not.

**Q. What do you mean by primary keys and foreign keys?**

In RDBMS, the association between the two tables is represented as Primary key and foreign key relationship. Whereas the primary key table is the close relative table and the foreign key table is the infant table.

**Q. Explain an outer join?**

An outer join is utilized when one wants to choose all the records from a port whether it has fulfilled the join criteria or not.

