

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

1Q.Explain in detail about ACID property and tell how it is helpful in software development?

Ans: The ACID model is one of the oldest and most important concepts of database theory. It sets forward four goals that every database management system must strive to achieve: atomicity, consistency, isolation and durability. No database that fails to meet any of these four goals can be considered reliable.

Let's take a moment to examine each one of these characteristics in detail:

Atomicity states that database modifications must follow an "all or nothing" rule. Each transaction is said to be "atomic." If one part of the transaction fails, the entire transaction fails. It is critical that the database management system maintain the atomic nature of transactions in spite of any DBMS, operating system or hardware failure.

Consistency states that only valid data will be written to the database. If, for some reason, a transaction is executed that violates the database's consistency rules, the entire transaction will be rolled back and the database will be restored to a state consistent with those rules. On the other hand, if a transaction successfully executes, it will take the database from one state that is consistent with the rules to another state that is also consistent with the rules.

Isolation requires that multiple transactions occurring at the same time not impact each other's execution. For example, if Joe issues a transaction against a database at the same time that Mary issues a different transaction, both transactions should operate on the database in an isolated manner. The database should either perform Joe's entire transaction before executing Mary's or vice-versa. This prevents Joe's transaction from reading intermediate data produced as a side effect of part of Mary's transaction that will not eventually be committed to the database. Note that the isolation property does not ensure which transaction will execute first, merely that they will not interfere with each other.

Durability ensures that any transaction committed to the database will not be lost. Durability is ensured through the use of database backups and transaction logs that facilitate the restoration of committed transactions in spite of any subsequent software or hardware failures.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

There are several different approaches to software development, much like the various views of political parties toward governing a country. Some take a more structured, engineering-based approach to developing business solutions, whereas others may take a more incremental approach, where software evolves as it is developed piece-by-piece. Most methodologies share some combination of the following stages of software development:

- **Analyzing the problem**
- **Market research**
- **Gathering requirements for the proposed business solution**
- **Devising a plan or design for the software-based solution**
- **Implementation (coding) of the software**
- **Testing the software**
- **Deployment**
- **Maintenance and bug fixing**

These stages are often referred to collectively as the software development lifecycle, or SDLC. Different approaches to software development may carry out these stages in different orders, or devote more or less time to different stages. The level of detail of the documentation produced at each stage of software development may also vary. These stages

may also be carried out in turn (a “waterfall” based approach), or they may be repeated over various cycles or iterations (a more “extreme” approach). The more extreme approach usually involves less time spent on planning and documentation, and more time spent on coding and development of automated tests. More “extreme” approaches also promote continuous testing throughout the development lifecycle, as well as having a working (or bug-free) product at all times. More structured or “waterfall” based approaches attempt to assess the majority of risks and develop a detailed plan for the software before implementation (coding) begins, and avoid significant design changes and re-coding in later stages of the software development lifecycle planning.

There are significant advantages and disadvantages to the various methodologies, and the best approach to solving a problem using software will often depend on the type of problem. If the problem is well understood and a solution can be effectively planned out ahead of time, the more

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

"waterfall" based approach may work the best. If, on the other hand, the problem is unique (at least to the development team) and the structure of the software solution cannot be easily envisioned, then a more "extreme" incremental approach may work best. A software development process is a structure imposed on the development of a software product. Synonyms include software life cycle and software process. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process.

2Q. Is it possible to write COM program using existing OLE technology? Write short note on it.

Ans: Yes, It is possible Com (Component object model) program using OLE(object linking embedding) technology write conclusion is given bellow:

OLE compound documents enable users working within a single application to manipulate data written in various formats and derived from multiple sources. For example, a user might insert into a word processing document a graph created in a second application and a sound object created in a third application. Activating the graph causes the second application to load its user interface, or at least that part containing tools necessary to edit the object. Activating the sound object causes the third application to play it. In both cases, a user is able to manipulate data from external sources from within the context of a single document.

OLE compound document technology rests on a foundation consisting of COM, structured storage, and uniform data transfer. As summarized below, each of these core technologies plays a critical role in OLE compound documents:

COM(Component object model): A compound document object is essentially a COM object that can be embedded in, or linked to, an existing document. As a COM object, a compound document object exposes the IUnknown interface, through which clients can obtain pointers to its other interfaces, including several, such as IOleObject, IOleLink, and IViewObject2, that provide special features unique to compound document objects.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Structured Storage: A compound document object must implement the **IPersistStorage** or, optionally, **IPersistStream** interfaces to manage its own storage. A container used to create compound documents must supply the **IStorage** interface, through which objects store and retrieve data. Containers almost always provide instances of **IStorage** obtained from OLE's Compound Files implementation. Containers must also use an object's **IPersistStorage** and/or **IPersistStream** interfaces.

Uniform Data Transfer: Applications that support compound documents must implement **IDataObject** because embedded objects and linked objects begin as data that has been transferred using special OLE clipboard formats, rather than standard Microsoft Windows clipboard formats. In other words, formatting data as an embedded or linked object is simply one more option provided by OLE's uniform data transfer model.

OLE's compound document technology benefits both software developers and users alike. Instead of feeling obligated to cram every conceivable feature into a single application, software developers are now free, if they like, to develop smaller, more focused applications that rely on other applications to supply additional features. In cases where a software developer decides to provide an application with capabilities beyond its core features, the developer can implement these additional services as separate DLLs, which are loaded into memory only when their services are required. Users benefit from smaller, faster, more capable software that they can mix and match as needed, manipulating all required components from within a single master document.

Interface	Comments/What it Means to Support the Interface
IOleObject	If the control requires communication with its client site for anything other than events (see IConnectionPointContainer), then IOleObject is a necessity. When implementing this interface, the control must also support the semantics of the following methods: SetHostNames() , Close() , EnumVerbs() , Update() , IsUpToDate() , GetUserClassID() , GetUserType() , GetMiscStatus() , and the Advise() , Unadvise() , and EnumAdvise()

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

	<p>methods that work in conjunction with a container's IAdviseSink implementation. A control implementing IOleObject must be able to handle IAdviseSink if the container provides one; a container may not, in which case a control ensures, of course, that it does not attempt to call a non-existent sink.</p>
IOleInPlaceObject	<p>Expresses the control's ability to be in-place activated and possibly UI activated. This interface means that the control has a user interface of some kind that can be activated, and IOleInPlaceActiveObject is supported as well. Required methods are GetWindow(), InPlaceActivate(), UIDeactivate(), SetObjectRects(), and ReactivateAndUndo(). Support for this interface requires support for IOleObject.</p>
IOleInPlaceActiveObject	<p>An in-place capable object that supports IOleInPlaceObject must also provide this interface as well, though the control itself doesn't necessarily implement the interface directly.</p>
IOleControl	<p>Expresses the control's ability and desire to deal with:</p> <ul style="list-style-type: none">(a) Mnemonics (GetControllInfo(), OnMnemonic() methods),(b) Ambient properties (OnAmbientPropertyChange()), and/or(c) Events that the control requires the container to handle (FreezeEvents()).

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Note that mnemonics are different than accelerators that are handled through IOleInPlaceActiveObject: mnemonics have associated UI and are active even when the control is not UI active. A control's support for mnemonics means that the control also knows how to use the container's

IOleControlSite::OnControlInfoChanged method. Because this requires the control to know the container's site, support for mnemonics also means support for IOleObject. In addition, knowledge of mnemonics requires in-place support and thus IOleInPlaceObject.

If a control uses any container-ambient properties, then it must also implement this interface to receive change notifications, as following the semantics of changes is required. Because ambient properties are only available through the container site's IDispatch, ambient property support means that the control supports IOleObject (to get the site) as well as being able to generate IDispatch::Invoke calls.

The FreezeEvents() method is necessary for controls that must know when a container is not going to handle an event; this is the only way for control to know this condition. If FreezeEvents() is only necessary in isolation, such that other IOleControl methods are not implemented, then IOleControl can stand alone without IOleObject or IOleInPlaceObject.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

IDataObject	<p>Indicates that the control can supply at least:</p> <p>(a) Graphical renderings of the control (CF_METAFILEPICT is the minimum if the control has any visuals at all) and/or</p> <p>(b) Property sets, if the control has any properties to provide.</p> <p>The methods GetData(), QueryGetData(), EnumFormatEtc(), DAdvise(), DUnadvise(), and EnumDAdvise() are required. Support for graphical formats other than CF_METAFILEPICT is optional.</p>
IViewObject2	<p>Indicates that the control has some interesting visuals when it is not in-place active. If implemented, a control must support the methods Draw(), GetAdvise(), SetAdvise(), and GetExtent().</p>
IDispatch	<p>Indicates that the control has either:</p> <p>(a) Custom methods, or</p> <p>(b) Custom properties that are both available via late-binding through IDispatch::Invoke.</p> <p>This also requires that the control provides type information through other IDispatch methods. A control may support multiple IDispatch implementations where only one is associated with IID_IDispatch, the</p>

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

	<p>others must have their own unique disinterface identifiers. A control is encouraged to supply dual interfaces for custom method and property access, but this is optional if methods and properties exist.</p>
ICollectionPointContainer	<p>Indicates that a control supports at least one outgoing interface, such as events or property change notifications. All methods of this interface must be implemented if this interface is available at all, including EnumConnectionPoints() which requires a separate object with ICollectionConnectionPoints().</p> <p>Support for ICollectionPointContainer means that the object also supports one or more related objects with ICollectionPoint that are available through ICollectionPointContainer methods. Each connection point object itself must implement the full ICollectionPoint interface, including EnumConnections(), which requires another enumerator object with the ICollectionConnections() interface.</p>

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION
MAHARAJA COLLEGE ARA

IProvideClassInfo[2]	Indicates that the object can provide its own coclass type information directly through IProvideClassInfo::GetClassInfo. If the control supports the later variation IProvideClassInfo2(), then it also indicates its ability to provide its primary source IID through IProvideClassInfo2::GetGUID. All methods of this interface must be implemented.
ISpecifyPropertyPages	Indicates that the control has property pages that it can display such that a container can coordinate this control's property pages with other control's pages when property pages are to be shown for a multi-control selection. All methods of this interface must be implemented when support exists.
IPerPropertyBrowsing	Indicates the control's ability to: (a) Provide a display string for a property, (b) Provide pre-defined strings and values for its properties and/or (c) Map a property dispID to a specific property page.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

	<p>Support for this interface means that support for properties through IDispatch as above is provided. If (c) is supported, then it also means that the object's property pages mapped through IPerPropertyBrowsing::MapPropertyToPage themselves implement IPropertyPage2 as opposed to the basic IPropertyPage interface.</p>
IPersistStream	See Persistence Interfaces section.
IPersistStreamInit	See Persistence Interfaces section.
IPersistMemory	See Persistence Interfaces section.
IPersistStorage	See Persistence Interfaces section.
IPersistMoniker	See Persistence Interfaces section.
IPersistPropertyBag	See Persistence Interfaces section.
IOleCache[2]	<p>Indicates support for container caching of control visuals. A control generally obtains caching support itself through the OLE function CreateDataCache(). Only controls with meaningful static content should choose to do this (although it is not required). If a control supports caching at all, it should simply aggregate the data cache and expose both IOleCache and IOleCache2 interfaces from the data cache. A control implementing IOleObject must be able to handle IAdviseSink if the container provides one; a container may not, in which case a control ensures, of course, that it does not attempt to call a non-existent sink.</p>
IExternalConnection	Indicates that the control supports external links to itself; that is, the control

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

	<p>is not marked with OLEMISC_CANTLINKINSIDE and supports IOleObject::SetMoniker and IOleObject::GetMoniker. A container will never query for this interface itself nor call it directly as calls are generated from inside OLE's remoting layer.</p>
IRunnableObject	<p>Indicates that the control differentiates being loaded from being running, as some in-process objects do.</p>

3Q. How clustering factor improve and help in COM programming?

Ans: The clustering factor is a number which represent the degree to which data is randomly distributed in a table.

In simple terms it is the number of "block switches" while reading a table using an index.

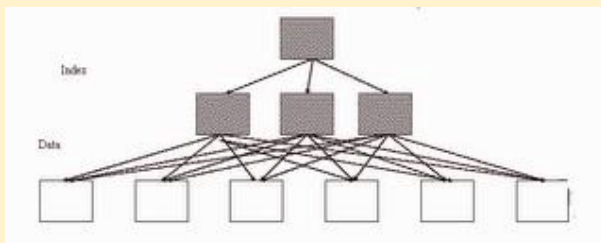


Figure: Bad clustering factor

the above diagram explains that how scatter the rows of the table are. The first index entry (from left of index) points to the first data block and second index entry points to second data block. So while making index range scan or full index scan, optimizer have to switch between blocks and have to revisit the same block more than once because rows are scatter. So the number of times optimizer will make these switches is actually termed as **"Clustering factor"**.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

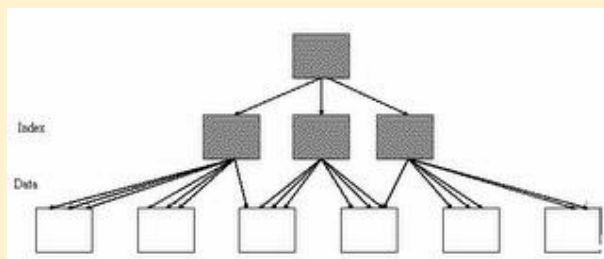


Figure: Good clustering factor

The above image represents "Good CF". In an event of index range scan, optimizer will not have to jump to next data block as most of the index entries points to same data block. This helps significantly in reducing the cost of your **SELECT** statements.

Clustering factor is stored in data dictionary and can be viewed from `dba_indexes` (or `user_indexes`)

```
SQL> create table sac as select * from all_objects;
```

Table created.

```
SQL> create index obj_id_indx on sac(object_id);
```

Index created.

```
SQL> select clustering_factor from user_indexes where  
index_name='OBJ_ID_INDX';
```

CLUSTERING_FACTOR

545

```
SQL> select count(*) from sac;
```

COUNT(*)

38956

```
SQL> select blocks from user_segments where  
segment_name='OBJ_ID_INDX';
```

BLOCKS

96

The above example shows that index has to jump 545 times to give you the full data had you performed full table scan using the index.

Note:

- A good CF is equal (or near) to the values of number of blocks of table.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

- A bad CF is equal (or near) to the number of rows of table.

Myth:

- Rebuilding of index can improve the CF.

Then how to improve the CF?

- To improve the CF, it's the table that must be rebuilt (and reordered).

- If table has multiple indexes, careful consideration needs to be given by which index to order table.

Clustering is a technique used to make several computers act as one larger machine, splitting tasks amongst them. They allow one to take several cheap stations, and combine them together to a larger system. It also allows for more redundancy for the system - if one machine in the cluster dies off, the other machines can cover up for it until the malfunctioning machine is repaired, and all this without bringing the whole system down. This type of setup is thus common in systems that must run 24 hours non-stop.

Clustering is often implemented in software, often using a protocol named PVM to communicate between the different machines. Examples for such systems are Beowulf, for Linux systems, or the clustering system by Tandem corporation.

4Q. What do you mean by persistence object?

Ans: Persistence simply means that we would like our application's data to outlive the applications process. In Java terms, we would like the state of (some of) our objects to live beyond the scope of the JVM so that the same state is available later.

A persistent object is an object that has been assigned a storage location in a federated database. When you commit the transaction in which you create a persistent object, that object's data is saved in the database; the object can then be accessed by other processes. A persistent object continues to exist beyond the duration of the process that creates it. In contrast, a transient object exists only within the memory of the process that creates it; when that process terminates, the transient object ceases to exist.

Making an Object Persistent

Only instances of [persistence-capable classes](#) can be persistent objects. Each application defines its own persistence-capable classes. In addition,

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Objectivity for Java includes persistence-capable classes for collections of persistent objects.

A persistent object can be:

- Given a **name** to facilitate retrieving the object from the database.
- **Related** to another persistent object.
- Added to a **persistent collection**.

If you perform any of these operations on a transient object, that object is made persistent. If you attempt to perform one of these operations on an object whose class is not persistence-capable, aNonPersistentClassException is thrown.

Container classes are themselves persistence-capable. A container is both a storage object and a persistent object; you can name a container, reference a container in a persistent field of another persistent object, add a container to a persistent collection, and establish a relationship from a persistent object to a container. You can define your own container classes if you need to create containers that have persistent data or relationships; however, most applications have no need to define their own container classes.

When you use the new operator to create an object of a persistence-capable class, the newly created object is transient. On the other hand, when you create an object by copying (using the persistent operation copy, not the Java clone method), an existing persistent object, the new copy is made persistent automatically. This section explains how to make a transient object persistent; "Copying a Persistent Object**" explains how to copy an object, creating a new object that is automatically persistent.**

You can make a transient object persistent only while a session is in a transaction; the newly persistent object belongs to that session. When you make an object persistent, the corresponding persistent object is created in the federated database. When you commit or checkpoint the transaction, the object is made visible to other clients. If you abort the transaction, the object in memory reverts to being transient and the new persistent object is removed from the federated database.

Working With a Persistent Object

A session can work with a persistent object once all the following conditions have been satisfied:

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

- The session is in a transaction.
- The session has a local representation of the persistent object. This condition is satisfied if you **make the object persistent** or **retrieve it from the database** while the session is in a transaction.
- The session has a **lock** on the object for the desired access. This condition is usually satisfied automatically. For example, the method that fetches the object's data also locks it.
- The session's local representation of the object has the object's most current persistent data. You must **fetch** the object's data during the transaction to obtain the most current data from the database.

If a particular operation does not require access to the object's data (the actual values of its field members), you need not fetch the data. In **Objectivity for Java**, no persistent operations require doing a fetch. For example, deleting, binding, looking up scope names, linking, and moving do not require a fetch prior to execution.

5Q.How DNA is associated with DCOM explain?

Ans: If you're developing applications to work with Microsoft® Windows® platforms, you've probably seen quite a bit of mention about Windows Distributed interNet Applications (DNA) and how it can help you create well-designed computer solutions in a distributed network environment. Even if you don't see a direct correlation between Windows DNA and your own development efforts, you might be surprised to find out just how versatile this architecture really is. Windows DNA is a programming model that can improve almost every programmer's work--it's not just for developers working on huge corporate applications. In fact, it can be utilized by *any* programmer developing for Windows today, whether you're writing a "clock" application, a high-end word processor, or factory automation software.

Windows DNA is often used to enable corporations to develop and deploy high-end distributed applications that integrate and expose services and solutions across their corporate network. But it also provides exciting potential for developers of applications that do not traditionally address the corporate market.

Expand Your Market

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Take, for example, vertical and horizontal market applications. Business applications are often thought of as being "vertical market," since they are created for a target market that is very specific to a particular industry or market segment. Oftentimes, that application has terrific functionality, but is so targeted toward a specific customer that it is useless to anybody else. Examples of vertical market applications include programs designed specifically for the health care industry, brokerage services, and, say, restaurant management. Compare this to "horizontal market" applications--software written to target a general task undertaken by a wide variety of users. Examples of horizontal applications include word processors, spreadsheets, terminal emulators, and web browsers.

Vertical market applications often contain the same functionality found in horizontal applications. For example, the vertical application might provide the spellchecker of a word processor, the mathematical formulas of a spreadsheet, and the mainframe connectivity of a terminal emulator. The key difference is that the vertical market application is written around a business logic that understands the needs and workflow unique to a particular industry.

Windows DNA allows for just this type of vertical/horizontal integration. One of its primary features is its use of Component Object Model (COM) components. COM components allow the business logic to be easily distributed across multiple servers. They also enable developers to create individual components that expose their specific and strategic capabilities. Application features can then be viewed as components that can be developed individually, and then integrated into a larger solution.

Don't Wait: *Integrate*

One of the easiest ways to integrate componentized applications is to utilize the scripting interfaces available on Windows. The best known method for adding functionality through scripting is to use Active Server Pages (ASP). This allows a simple component to be installed onto a web server, integrating the functionality of the component into the web page through server side scripting. The component itself can be as simple as a hit counter, or as complex as a robust shopping cart component.

A scriptable component can also be utilized by Windows Script code running outside of the browser. With the recently released Windows Script Host (which ships with Windows 98 and is freely downloadable from [MSDN's Scripting site](#) for Windows 95 and Windows NT® 4.0), it's easy to develop script code that runs at the system level outside of the security confines of a web browser. This enables you to easily combine

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

the functionality of multiple applications and components into a single seamless solution.

Let's take spell-checking functionality as an example. Such functionality can be exposed in two different ways. One way would be for a large high-end word processor to allow scriptable access to its internal spell-checker. Another way would be to develop a stand-alone spell-checker component specifically designed for scriptable access. This level of targeted component development allows for high flexibility and a highly customized solution.

So, if your company specializes in spell-checking technology, but not in typesetting or graphical layout, the best approach would be to develop a specialized component that leverages your strengths. This allows you to participate in a much larger solution, in a manner that might otherwise have not been available to you.

Such an approach also makes it easier to build large vertical applications out of a selection of quality components. You can use the traditional approach of using Visual Basic® to build custom applications, and you can integrate web page access to those features and functionality where appropriate. Since the underlying scripting access is the same between Visual Basic and Windows Script, training for one also covers most of the training necessary for the other.

If you're a developer who thought Windows DNA didn't really apply to you, I hope I've convinced you otherwise. Of course, Windows DNA contains many other features and benefits; I've only touched upon a small sub-set here. If you take the time to delve further, I'm sure you'll uncover many more additional ways to let Windows DNA help you in your development efforts.

6Q. what is message queuing and how it is helpful in window environment and when it is used?

Ans: Messages Queuing: Well, you guessed it. These are the kind of queues that store plain old application messages. They are the targets of most message, and place from which standard messages are received. Message queues are usually created by application; however some applications might require that the MSMQ administrator create specific queues.

Application message queues can be public or private. Public queues are registered in the active directory (or an interim substitute on Windows NT 4) and are available throughout the MSMQ Enterprise. Private queues are

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

visible only on the MSMQ machine on which they were created.

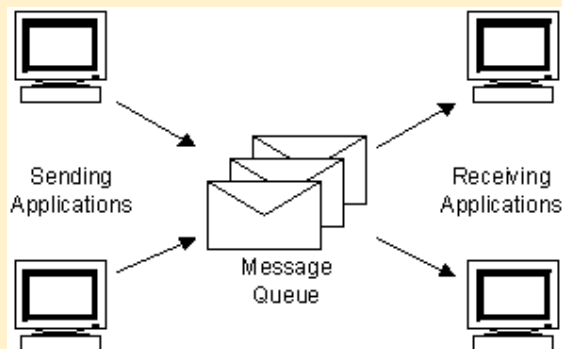
Application can pass the names of private queues to remote application in order to support response message.

Unlike MS-DOS-based applications, Windows-based applications are event-driven. They do not make explicit function calls (such as C run-time library calls) to obtain input. Instead, they wait for the system to pass input to them.

The system passes all input for an application to the various windows in the application. Each window has a function, called a window procedure, that the system calls whenever it has input for the window. The window procedure processes the input and returns control to the system. For more information about window procedures, see [Window Procedures](#).

If a top-level window stops responding to messages for more than several seconds, the system considers the window to be not responding. In this case, the system hides the window and replaces it with a ghost window that has the same Z order, location, size, and visual attributes. This allows the user to move it, resize it, or even close the application. However, these are the only actions available because the application is actually not responding. When in the debugger mode, the system does not generate a ghost window.

Purpose of Message Queues: Message Queuing (MSMQ) technology enables applications running at different times to communicate across heterogeneous networks and systems that may be temporarily offline. Applications send messages to queues and read messages from queues. The following illustration shows how a queue can hold messages that are generated by multiple sending applications and read by multiple receiving applications.



Where Applicable

Message Queuing provides guaranteed message delivery, efficient routing, security, and priority-based messaging.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

It can be used to implement solutions to both asynchronous and synchronous scenarios requiring high performance. The following list shows several places where Message Queuing can be used.

- **Mission-critical financial services:** for example, electronic commerce.
- **Embedded and hand-held applications:** for example, underlying communications to and from embedded devices that route baggage through airports by means of an automatic baggage system.
- **Outside sales:** for example, sales automation applications for traveling sales representatives.
- **Workflow:** Message Queuing makes it easy to create a workflow that updates each system. A typical design pattern is to implement an agent to interact with each system. Using a workflow-agent architecture also minimizes the impact of changes in one system on the other systems. With Message Queuing, the loose coupling between systems makes upgrading individual systems simpler.

Developer Audience

Message queuing applications can be developed using C++ APIs or COM objects. Applications can be built in any of the popular development environments: for example, Microsoft® Visual Basic®, Visual Basic® Scripting Edition, Visual C++®, Visual Studio® .NET, Borland Delphi, and Power soft PowerBuilder. This allows applications to be developed for the Internet as well, including both server-side (Internet Information Server) and client-side (Internet Explorer) applications. The .NET Framework offers a set of managed Message Queuing objects.

Windows Messages:

The system passes input to a window procedure in the form of a *message*. Messages are generated by both the system and applications. The system generates a message at each input event—for example, when the user types, moves the mouse, or clicks a control such as a scroll bar. The system also generates messages in response to changes in the system brought about by an application, such as when an application changes the pool of system font resources or resizes one of its windows. An application can generate messages to direct its own windows to perform tasks or to communicate with windows in other applications. The system sends a message to a window procedure with a set of four parameters: a window handle, a message identifier, and two values called *message parameters*. The *window handle* identifies the window for

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

which the message is intended. The system uses it to determine which window procedure should receive the message.

A *message identifier* is a named constant that identifies the purpose of a message. When a window procedure receives a message, it uses a message identifier to determine how to process the message. For example, the message identifier WM_PAINT tells the window procedure that the window's client area has changed and must be repainted. Message parameters specify data or the location of data used by a window procedure when processing a message. The meaning and value of the message parameters depend on the message. A message parameter can contain an integer, packed bit flags, a pointer to a structure containing additional data, and so on. When a message does not use message parameters, they are typically set to NULL. A window procedure must check the message identifier to determine how to interpret the message parameters.

7Q.What is security is performed in component object model?

Ans: Security in COM is firmly based on the security provided by Windows and the underlying RPC security mechanisms. COM security relies on *authentication* (the process of verifying a caller's identity) and *authorization* (the process of determining whether a caller is authorized to do what it is asking to do). There are two main types of security in COM: *activation security* and *call security*. Activation security determines whether a client can launch a server at all. After a server has been launched, you can use call security to control access to a server's objects.

In this security model, servers manage and help protect objects, clients get access to objects through servers, and servers can attempt access while impersonating the client.

The system implements the Kerberos v5 authentication protocol and the Schannel security package. It also includes features such as delegate-level impersonation, mutual authentication, the ability to set authentication levels for an App ID in the registry, and cloaking. Using COM security, you can implement objects that can perform privileged operations without compromising security.

Because there is a wide range of COM security features available, it is helpful to initially determine what kind of security your application needs. For most applications, setting an acceptable level of security can be a

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

painless process, but you can also use COM security to support very complex security scenarios.

Activation security (also called launch security) helps control who can launch a server. Activation security is automatically applied by the service control manager (SCM) of a particular computer. Upon receipt of a request from a client to activate an object (as described in [Instance Creation Helper Functions](#)), the SCM checks the request against activation-security information stored within its registry. (Activation security is also checked for same-computer activations.)

When determining the identity of the client, activation examines the cloaking flag set in the client's call to `CoInitializeSecurity`. If the cloaking flag is set (for either dynamic or static cloaking), the thread token is used, if present, to determine the identity of the client. If no cloaking is set, the process token is used instead of the thread token.

The authentication level controls how much security a client or server wants from its SSP. The authentication level is set by passing an appropriate `RPC_C_AUTHN_LEVEL_xxx` value to `CoInitializeSecurity` or `CoSetProxyBlanket` through the `dwAuthnLevel` parameter. The authentication levels from the client and server are compared during the handshake, and the higher level security protection setting is used for the connection.

The different authentication levels are described as follows, from lowest level security protection to highest:

None (RPC_C_AUTHN_LEVEL_NONE)

No authentication is performed during the communication between client and server. All security settings are ignored. This authentication level can be set only if the authentication service level is `RPC_C_AUTHN_NONE`.

Default (RPC_C_AUTHN_LEVEL_DEFAULT)

COM chooses the authentication level by using its normal security blanket negotiation. It will never choose an authentication level of None.

Connect (RPC_C_AUTHN_LEVEL_CONNECT)

The normal authentication handshake occurs between the client and server, and a session key is established but that key is never used for communication between the client and server. All communication after the handshake is no secure.

Call (RPC_C_AUTHN_LEVEL_CALL)

Only the headers of the beginning of each call are signed. The rest of the data exchanged between the client and server is neither signed nor encrypted. Most SSPs do not support this authentication level and silently promote it to Packet.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

8Q. Write a simple code in C++ or any other language, the demonstrate the use of COM component?

Ans: Option Explicit

Private mdblCelsius As Double

Private mdblFahrenheit As Double

Public Event BelowFreezing()

Public Event AboveBoiling()

Public Property Get Celsius() As Double

Celsius = mdblCelsius

End Property

Public Property Let Celsius(NewTemperature As Double)

mdblCelsius = NewTemperature

mdblFahrenheit = ((NewTemperature * 9) / 5) + 32

If mdblCelsius < 0 Then

RaiseEvent BelowFreezing

End If

If mdblCelsius > 100 Then

RaiseEvent AboveBoiling

End If

End Property

Public Property Get Fahrenheit() As Double

Fahrenheit = mdblFahrenheit

End Property

Public Property Let Fahrenheit(NewTemperature As Double)

mdblFahrenheit = NewTemperature

mdblCelsius = ((NewTemperature - 32) * 5) / 9

If mdblFahrenheit < 32 Then

RaiseEvent BelowFreezing

End If

If mdblFahrenheit > 212 Then

RaiseEvent AboveBoiling

End If

End Property

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Public Function GetCelsius() As Double

GetCelsius = mdblCelsius

End Function

Public Function GetFahrenheit() As Double

GetFahrenheit = mdblFahrenheit

End Function

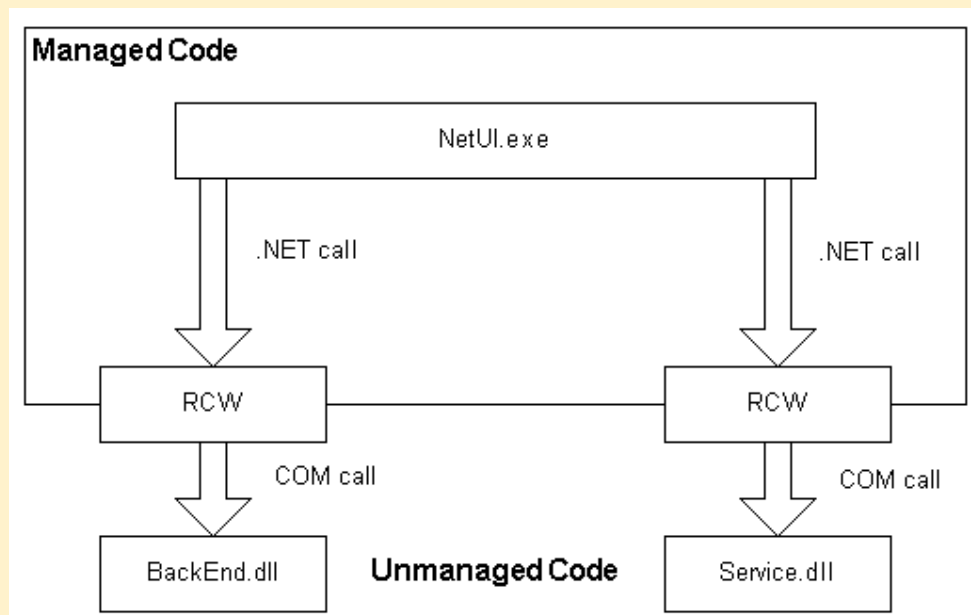
Private Sub Class_Initialize()

mdblCelsius = 0

mdblFahrenheit = 32

End Sub

Manage and Unmanaged coding in Component object model:



9Q. What is Multi-tier architecture and how it is different from three tier architecture?

Ans: Multi-Tier architecture: In software engineering, multi-tier architecture (often referred to as n-tier architecture) is a client-server architecture in which presentation, application processing, and data management functions are logically separated. For example, an application that uses middleware to service data requests between a user and a database employs multi-tier architecture. The most widespread use of multi-tier architecture is the three-tier architecture.

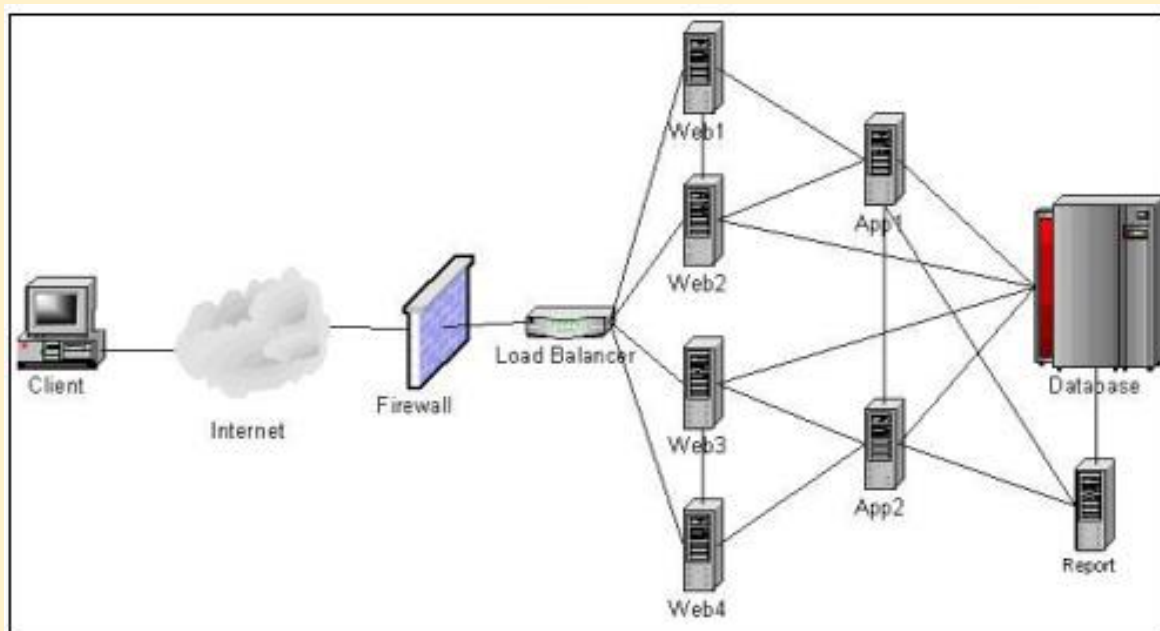
INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

N-tier application architecture provides a model by which developers can create flexible and reusable applications. By segregating an application into tiers, developers acquire the option of modifying or adding a specific layer, instead of reworking the entire application. Three-tier architectures typically comprise a presentation tier, a business or data access tier, and a data tier.

While the concepts of layer and tier are often used interchangeably, one fairly common point of view is that there is indeed a difference. This view holds that a layer is a logical structuring mechanism for the elements that make up the software solution, while a tier is a physical structuring mechanism for the system infrastructure.^{[1] [2]}

There have been growing concerns in recent years that many organizations are facing an excessive number of layers in their multi-layered architecture. These concerns stem from sprawling application architectures that are not well designed or managed, in which development teams create an ever-growing number of "wrapper" layers that compromise maintainability. The resulting architecture resembles a [Rube Goldberg Machine](#) that scares organizations from solving the root cause of the sprawling layers, resulting in the creation of more layers.



Three tier architecture: Three-tier^[3] is a [client-server architecture](#) in which the [user interface](#), [functional process logic](#) ("business

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

rules"), [computer data storage](#) and [data access](#) are developed and maintained as independent [modules](#), most often on separate [platforms](#). It was developed by [John J. Donovan](#) in Open Environment Corporation (OEC), a tools company he founded in [Cambridge, Massachusetts](#).

The three-tier model is a [software architecture](#) and a [software design pattern](#).

Apart from the usual advantages of modular [software](#) with well-defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently in response to changes in requirements or [technology](#). For example, a change of [operating system](#) in the presentation tier would only affect the user interface code.

Typically, the user interface runs on a desktop [PC](#) or [workstation](#) and uses a standard [graphical user interface](#), functional process logic may consist of one or more separate modules running on a workstation or [application server](#), and an [RDBMS](#) on a [database server](#) or [mainframe](#) contains the computer data storage logic. The middle tier may be multi-tiered itself (in which case the overall architecture is called an "n-tier architecture").

Three-tier architecture has the following three tiers:

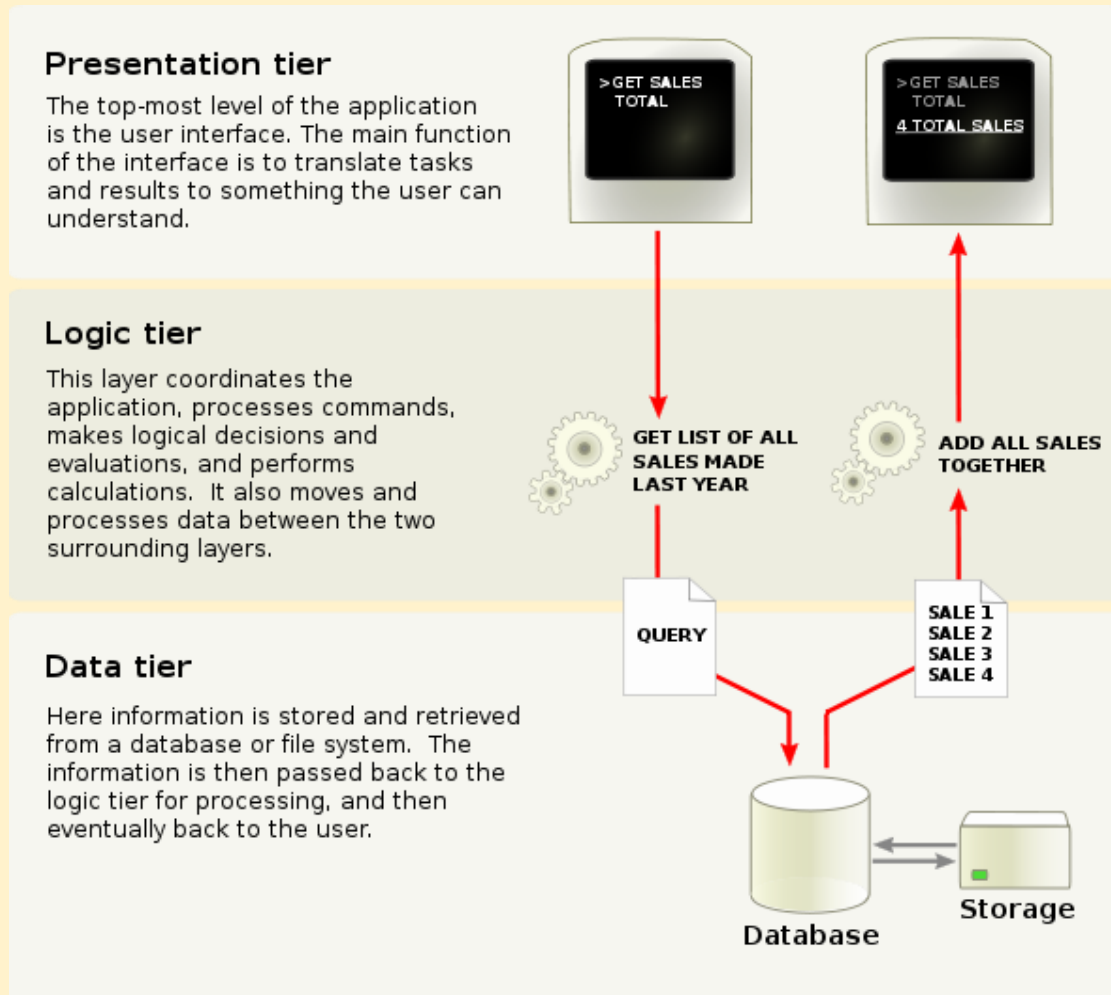
Presentation tier: This is the topmost level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing, and shopping cart contents. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network.

Application tier : ([business logic](#), logic tier, data access tier, or middle tier)
The logic tier is pulled out from the presentation tier and, as its own layer, it controls an application's functionality by performing detailed processing.

Data tier: This tier consists of database servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data on its own tier also improves scalability and performance.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA



Q10. Describe briefly about the sharing and scalability of COM and DCOM?

Ans: COM(Component object model): Component Object Model (COM) specifies an architecture, a binary standard, and a supporting infrastructure for building, using, and evolving component-based applications. It extends the benefits of object-oriented programming such as encapsulation, polymorphism, and software reuse to a dynamic and cross-process setting. Distributed COM (DCOM) is the distributed extension of COM. It specifies the additional infrastructure that is required to further extend the benefits to networked environments.

Distributed computing is becoming a mainstream due to the advance in high-speed networking and the explosive growth of the Internet. Object-oriented programming has become a dominating programming paradigm

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

for developing reusable software. Distributed objects combine the two trends and are becoming increasingly popular. More and more software systems are being build as distributed object applications and they often share a number of common goals. The main objective of this paper is to identify the main features of COM/DCOM, which greatly facilitates achieving the common goals. Such features include the separation of interfaces and implementations, support for objects with multiple interfaces, language neutrality, run-time binary software reuse, location transparency, architecture for extensibility, support for indirection, approach to versioning, and different styles of server lifetime management. The argument is that, by using COM/DCOM as a platform for building distributed object applications, researchers and developers can concentrate on important issues specific to their applications without having to devote a significant portion of their efforts to building the supporting infrastructure.

COM is a platform-independent, distributed, object-oriented system for creating binary software components that can interact. COM is the foundation technology for Microsoft's OLE (compound documents) and ActiveX (Internet-enabled components) technologies.

COM Basics:

Object model :The separation of interface and implementation is at the core of COM. An interface is a collection of functionally related abstract methods, and is identified by a 128-bit globally unique identifier (GUID) called the interface ID (IID). In contrast, an object class is a concrete implementation of one or more interfaces, and is also identified by a GUID called the class ID (CLSID). The use of GUIDs allows programmers to independently generate unique IDs without requiring central authority. An object instance (or object) is an instantiation of some object class. An object server is a dynamic link library (DLL) or an executable (EXE) capable of creating object instances of potentially multiple classes. A client is a process that invokes methods of an object.

Binary interface standard :COM specifies a binary standard for interfaces to ensure dynamic interoperability of binary objects possibly built using different programming languages. Specifically, any COM interface must satisfy two requirements. First, its instantiation must follow a standard memory layout, which is the same as the C++ virtual function table [Box98]. In other words, a COM interface pointer is a pointer to a pointer that points to an array of virtual function pointers. Second, any COM interface must inherit from the IUnknown interface so that its first three methods are (1) Query Interface() for navigating between interfaces of the

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

same object instance, (2) **AddRef()** for incrementing reference counts, and (3) **Release()** for decrementing reference counts.

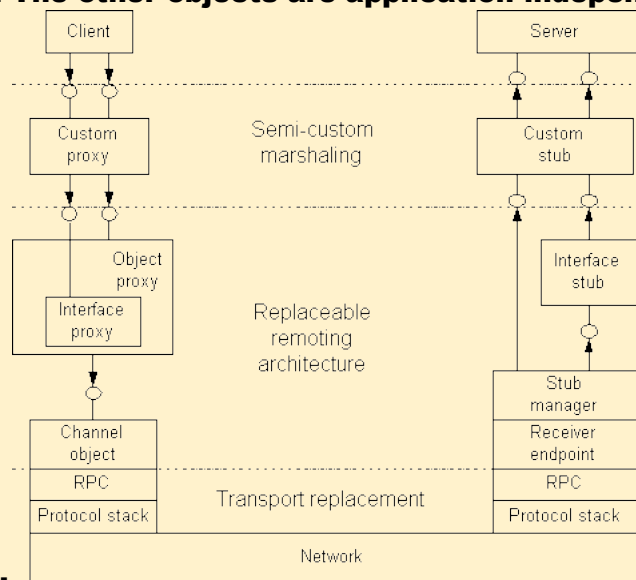
Programming model: A typical client/server interaction in COM goes like this: client starts the activation phase by calling **Co Create Instance()** with the **CLSID** of the requested object and the **IID** of the requested interface. It gets back an interface pointer from the call. Upon returning the interface pointer, the object calls **AddRef()** on itself. In the method invocation phase, the client invokes methods of the interface through the pointer as if the object resides in its own address space. When the client needs to call methods of another interface of the same object, it calls **QueryInterface()** on the current interface and specifies the **IID** of the second interface. Once it gets back a pointer to the second interface, it can invoke methods as usual. When the client finishes using either interface pointer, it calls **Release()** on the pointer.

Remoting architecture: We use the term remoting architecture [Chung97] to refer to the entire infrastructure that connects COM clients to out-of-process server objects. (See Figure 1.) The standard remoting architecture includes, among other things, (1) object proxies that act as the client-side representatives of server objects and connect directly to the client; (2) interface proxies that perform client-side data marshaling and are aggregated into object proxies; (3) client-side channel objects that use remote procedure calls (RPCs) to forward marshaled calls; (4) server-side endpoints that receive RPC requests; (5) server-side stub manager that dispatches calls to appropriate interface stubs; (6) interface stubs that perform server-side data marshaling and make actual calls on the objects; and (7) standard marshaler that marshals interface pointers into object references on the server side and unmarshals the object references on the client side. Note that interface proxies and stubs are application-specific and are generated by running an Interface Definition Language (IDL) compiler on application-supplied IDL

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

files. The other objects are application-independent and are provided by



COM.

Purpose: COM is a platform-independent, distributed, object-oriented system for creating binary software components that can interact. COM is the foundation technology for Microsoft's OLE (compound documents) and ActiveX (Internet-enabled components) technologies.

Distributed components object model: Distributed Component Object Model (DCOM) is a proprietary Microsoft technology for communication among software components distributed across networked computers. DCOM, which originally was called "Network OLE", extends Microsoft's COM, and provides the communication substrate under Microsoft's COM+ application server infrastructure. It has been deprecated in favor of the Microsoft .NET Remoting, a part of their .NET Framework.^{[1][2]}

The addition of the "D" to COM was due to extensive use of DCE/RPC (Distributed Computing Environment/Remote Procedure Calls) – more specifically Microsoft's enhanced version, known as MSRPC.

In terms of the extensions it added to COM, DCOM had to solve the problems of

- **Marshaling – serializing and desterializing the arguments and return values of method calls "over the wire".**

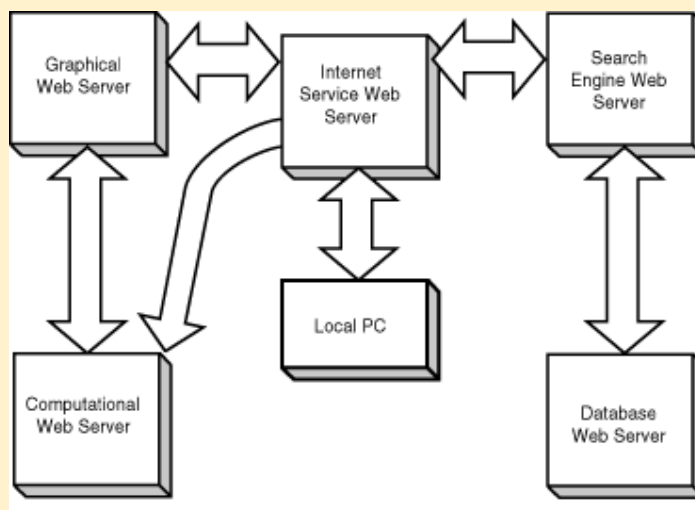
INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

- **Distributed garbage collection** – ensuring that references held by clients of interfaces are released when, for example, the client process crashed, or the network connection was lost.
- **Aggregating hundreds or potentially tens of thousands of references to objects held by clients of interfaces at a single host, into a single "ping" function, in order to minimise bandwidth utilisation.**

One of the key factors in solving these problems is the use of DCE/RPC as the underlying RPC mechanism behind DCOM. DCE/RPC has strictly defined rules regarding marshalling and who is responsible for freeing memory.

DCOM was a major competitor to CORBA. Proponents of both of these technologies saw them as one day becoming the model for code and service-reuse over the Internet. However, the difficulties involved in getting either of these technologies to work over Internet firewalls, and on unknown and insecure machines, meant that normal HTTP requests in combination with web browsers won out over both of them. Microsoft, at one point, attempted and failed to head this off by adding an extra http transport to DCE/RPC called ncacn_http (Network Computing Architecture, Connection-based, over HTTP). This was later resurrected to support a Microsoft Exchange 2003 connection over HTTP.



Q11.Explain persistence object with example?

Ans: : Persistence simply means that we would like our application's data to outlive the applications process. In Java terms, we would like the state

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

of (some of) our objects to live beyond the scope of the JVM so that the same state is available later.

A persistent object is an object that has been assigned a storage location in a federated database. When you commit the transaction in which you create a persistent object, that object's data is saved in the database; the object can then be accessed by other processes. A persistent object continues to exist beyond the duration of the process that creates it. In contrast, a transient object exists only within the memory of the process that creates it; when that process terminates, the transient object ceases to exist.

Making an Object Persistent

Only instances of persistence-capable classes can be persistent objects. Each application defines its own persistence-capable classes. In addition, Objectivity for Java includes persistence-capable classes for collections of persistent objects.

A persistent object can be:

- Given a name to facilitate retrieving the object from the database.
- Related to another persistent object.
- Added to a persistent collection.

If you perform any of these operations on a transient object, that object is made persistent. If you attempt to perform one of these operations on an object whose class is not persistence-capable, a `NonPersistentClassException` is thrown.

Container classes are themselves persistence-capable. A container is both a storage object and a persistent object; you can name a container, reference a container in a persistent field of another persistent object, add a container to a persistent collection, and establish a relationship from a persistent object to a container. You can define your own container classes if you need to create containers that have persistent data or relationships; however, most applications have no need to define their own container classes.

When you use the new operator to create an object of a persistence-capable class, the newly created object is transient. On the other hand, when you create an object by copying (using the persistent operation `copy`, not the Java clone method), an existing persistent object, the new copy is made persistent automatically. This section explains how to make a transient object persistent; "Copying a Persistent

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Object" explains how to copy an object, creating a new object that is automatically persistent.

You can make a transient object persistent only while a session is in a transaction; the newly persistent object belongs to that session. When you make an object persistent, the corresponding persistent object is created in the federated database. When you commit or checkpoint the transaction, the object is made visible to other clients. If you abort the transaction, the object in memory reverts to being transient and the new persistent object is removed from the federated database.

Working With a Persistent Object

A session can work with a persistent object once all the following conditions have been satisfied:

- **The session is in a transaction.**
- **The session has a local representation of the persistent object. This condition is satisfied if you make the object persistent or retrieve it from the database while the session is in a transaction.**
- **The session has a lock on the object for the desired access. This condition is usually satisfied automatically. For example, the method that fetches the object's data also locks it.**
- **The session's local representation of the object has the object's most current persistent data. You must fetch the object's data during the transaction to obtain the most current data from the database.**

If a particular operation does not require access to the object's data (the actual values of its field members), you need not fetch the data. In Objectivity for Java, no persistent operations require doing a fetch. For example, deleting, binding, looking up scope names, linking, and moving do not require a fetch prior to execution.

Q12. What is message queuing ? Explain its use in windows environment?

Ans: Messages Queuing: Well, you guessed it. These are the kind of queues that store plain old application messages. They are the targets of most message, and place from which standard messages are received. Message queues are usually created by application; however some applications might require that the MSMQ administrator create specific queues.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Application message queues can be public or private. Public queues are registered in the active directory (or an interim substitute on Windows NT 4) and are available throughout the MSMQ Enterprise. Private queues are visible only on the MSMQ machine on which they were created.

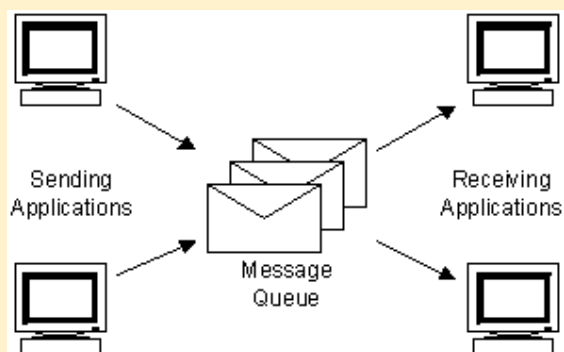
Application can pass the names of private queues to remote application in order to support response message.

Unlike MS-DOS-based applications, Windows-based applications are event-driven. They do not make explicit function calls (such as C run-time library calls) to obtain input. Instead, they wait for the system to pass input to them.

The system passes all input for an application to the various windows in the application. Each window has a function, called a window procedure, that the system calls whenever it has input for the window. The window procedure processes the input and returns control to the system. For more information about window procedures, see Window Procedures.

If a top-level window stops responding to messages for more than several seconds, the system considers the window to be not responding. In this case, the system hides the window and replaces it with a ghost window that has the same Z order, location, size, and visual attributes. This allows the user to move it, resize it, or even close the application. However, these are the only actions available because the application is actually not responding. When in the debugger mode, the system does not generate a ghost window.

Purpose of Message Queues: Message Queuing (MSMQ) technology enables applications running at different times to communicate across heterogeneous networks and systems that may be temporarily offline. Applications send messages to queues and read messages from queues. The following illustration shows how a queue can hold messages that are generated by multiple sending applications and read by multiple receiving applications.



Where Applicable

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Message Queuing provides guaranteed message delivery, efficient routing, security, and priority-based messaging.

It can be used to implement solutions to both asynchronous and synchronous scenarios requiring high performance. The following list shows several places where Message Queuing can be used.

- **Mission-critical financial services: for example, electronic commerce.**
- **Embedded and hand-held applications: for example, underlying communications to and from embedded devices that route baggage through airports by means of an automatic baggage system.**
- **Outside sales: for example, sales automation applications for traveling sales representatives.**
- **Workflow: Message Queuing makes it easy to create a workflow that updates each system. A typical design pattern is to implement an agent to interact with each system. Using a workflow-agent architecture also minimizes the impact of changes in one system on the other systems. With Message Queuing, the loose coupling between systems makes upgrading individual systems simpler.**

Developer Audience

Message queuing applications can be developed using C++ APIs or COM objects. Applications can be built in any of the popular development environments: for example, Microsoft® Visual Basic®, Visual Basic® Scripting Edition, Visual C++®, Visual Studio® .NET, Borland Delphi, and Power soft PowerBuilder. This allows applications to be developed for the Internet as well, including both server-side (Internet Information Server) and client-side (Internet Explorer) applications. The .NET Framework offers a set of managed Message Queuing objects.

Windows Messages:

The system passes input to a window procedure in the form of a message. Messages are generated by both the system and applications. The system generates a message at each input event—for example, when the user types, moves the mouse, or clicks a control such as a scroll bar. The system also generates messages in response to changes in the system brought about by an application, such as when an application changes the pool of system font resources or resizes one of its windows. An application can generate messages to direct its own windows to perform tasks or to communicate with windows in other applications.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

The system sends a message to a window procedure with a set of four parameters: a window handle, a message identifier, and two values called message parameters. The window handle identifies the window for which the message is intended. The system uses it to determine which window procedure should receive the message.

A message identifier is a named constant that identifies the purpose of a message. When a window procedure receives a message, it uses a message identifier to determine how to process the message. For example, the message identifier WM_PAINT tells the window procedure that the window's client area has changed and must be repainted. Message parameters specify data or the location of data used by a window procedure when processing a message. The meaning and value of the message parameters depend on the message. A message parameter can contain an integer, packed bit flags, a pointer to a structure containing additional data, and so on. When a message does not use message parameters, they are typically set to NULL. A window procedure must check the message identifier to determine how to interpret the message parameters.

Q13. What is Three tier architecture? Explain with examples?

Ans: In software engineering, multi-tier architecture (often referred to as n-tier architecture) is a client-server architecture in which presentation, application processing, and data management functions are logically separated. For example, an application that uses middleware to service data requests between a user and a database employs multi-tier architecture. The most widespread use of multi-tier architecture is the three-tier architecture.

N-tier application architecture provides a model by which developers can create flexible and reusable applications. By segregating an application into tiers, developers acquire the option of modifying or adding a specific layer, instead of reworking the entire application. Three-tier architectures typically comprise a presentation tier, a business or data access tier, and a data tier.

While the concepts of layer and tier are often used interchangeably, one fairly common point of view is that there is indeed a difference. This view holds that a layer is a logical structuring mechanism for the elements that make up the software solution, while a tier is a physical structuring mechanism for the system infrastructure.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

There have been growing concerns in recent years that many organizations are facing an excessive number of layers in their multi-layered architecture. These concerns stem from sprawling application architectures that are not well designed or managed, in which development teams create an ever-growing number of "wrapper" layers that compromise maintainability. The resulting architecture resembles a Rube Goldberg Machine that scares organizations from solving the root cause of the sprawling layers, resulting in the creation of more layers.

Three-tier³¹ is a client-server architecture in which the user interface, functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms. It was developed by John J. Donovan in Open Environment Corporation (OEC), a tools company he founded in Cambridge, Massachusetts.

The three-tier model is a software architecture and a software design pattern.

Apart from the usual advantages of modular software with well-defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently in response to changes in requirements or technology. For example, a change of operating system in the presentation tier would only affect the user interface code.

Typically, the user interface runs on a desktop PC or workstation and uses a standard graphical user interface, functional process logic may consist of one or more separate modules running on a workstation or application server, and an RDBMS on a database server or mainframe contains the computer data storage logic. The middle tier may be multi-tiered itself (in which case the overall architecture is called an "n-tier architecture").

Three-tier architecture has the following three tiers:

Presentation tier

This is the topmost level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing, and shopping cart contents. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network.

Application tier (business logic, logic tier, data access tier, or middle tier)

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

The logic tier is pulled out from the presentation tier and, as its own layer, it controls an application's functionality by performing detailed processing.

Data tier

This tier consists of database servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data on its own tier also improves scalability and performance.

Web development usage

In the web development field, three-tier is often used to refer to websites, commonly electronic commerce websites, which are built using three tiers:

- 1. A front-end web server serving static content, and potentially some cached dynamic content. In web based application, Front End is the content rendered by the browser. The content may be static or generated dynamically.**
- 2. A middle dynamic content processing and generation level application server, for example Ruby on Rails, Java EE, ASP.NET, PHP, ColdFusion, Perl platform.**
- 3. A back-end database or data store, comprising both data sets and the database management system or RDBMS software that manages and provides access to the data.**

Q14. Explain shorts notes on COM and DCOM? Also differentiate between the on the basis of implementation?

Ans: : COM(Component object model):

Component Object Model (COM) specifies an architecture, a binary standard, and a supporting infrastructure for building, using, and evolving component-based applications. It extends the benefits of object-oriented programming such as encapsulation, polymorphism, and software reuse to a dynamic and cross-process setting. Distributed COM (DCOM) is the distributed extension of COM. It specifies the additional infrastructure that is required to further extend the benefits to networked environments.

Distributed computing is becoming a mainstream due to the advance in high-speed networking and the explosive growth of the Internet. Object-oriented programming has become a dominating programming paradigm

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

for developing reusable software. Distributed objects combine the two trends and are becoming increasingly popular. More and more software systems are being build as distributed object applications and they often share a number of common goals. The main objective of this paper is to identify the main features of COM/DCOM, which greatly facilitates achieving the common goals. Such features include the separation of interfaces and implementations, support for objects with multiple interfaces, language neutrality, run-time binary software reuse, location transparency, architecture for extensibility, support for indirection, approach to versioning, and different styles of server lifetime management. The argument is that, by using COM/DCOM as a platform for building distributed object applications, researchers and developers can concentrate on important issues specific to their applications without having to devote a significant portion of their efforts to building the supporting infrastructure.

COM is a platform-independent, distributed, object-oriented system for creating binary software components that can interact. COM is the foundation technology for Microsoft's OLE (compound documents) and ActiveX (Internet-enabled components) technologies.

COM Basics

Object model

The separation of interface and implementation is at the core of COM. An interface is a collection of functionally related abstract methods, and is identified by a 128-bit globally unique identifier (GUID) called the interface ID (IID). In contrast, an object class is a concrete implementation of one or more interfaces, and is also identified by a GUID called the class ID (CLSID). The use of GUIDs allows programmers to independently generate unique IDs without requiring central authority. An object instance (or object) is an instantiation of some object class. An object server is a dynamic link library (DLL) or an executable (EXE) capable of creating object instances of potentially multiple classes. A client is a process that invokes methods of an object.

Binary interface standard

COM specifies a binary standard for interfaces to ensure dynamic interoperability of binary objects possibly built using different programming languages. Specifically, any COM interface must satisfy two requirements. First, its instantiation must follow a standard memory layout, which is the same as the C++ virtual function table [Box98]. In other words, a COM interface pointer is a pointer to a pointer that points

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

to an array of virtual function pointers. Second, any COM interface must inherit from the IUnknown interface so that its first three methods are (1) QueryInterface() for navigating between interfaces of the same object instance, (2) AddRef() for incrementing reference counts, and (3) Release() for decrementing reference counts.

Programming model

A typical client/server interaction in COM goes like this: client starts the activation phase by calling CoCreateInstance() with the CLSID of the requested object and the IID of the requested interface. It gets back an interface pointer from the call. Upon returning the interface pointer, the object calls AddRef() on itself. In the method invocation phase, the client invokes methods of the interface through the pointer as if the object resides in its own address space. When the client needs to call methods of another interface of the same object, it calls QueryInterface() on the current interface and specifies the IID of the second interface. Once it gets back a pointer to the second interface, it can invoke methods as usual. When the client finishes using either interface pointer, it calls Release() on the pointer.

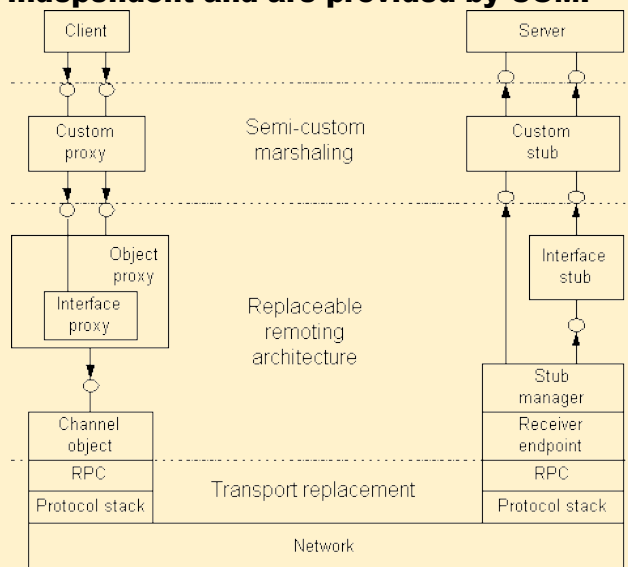
Remoting architecture

We use the term remoting architecture [Chung97] to refer to the entire infrastructure that connects COM clients to out-of-process server objects. (See Figure 1.) The standard remoting architecture includes, among other things, (1) object proxies that act as the client-side representatives of server objects and connect directly to the client; (2) interface proxies that perform client-side data marshaling and are aggregated into object proxies; (3) client-side channel objects that use remote procedure calls (RPCs) to forward marshaled calls; (4) server-side endpoints that receive RPC requests; (5) server-side stub manager that dispatches calls to appropriate interface stubs; (6) interface stubs that perform server-side data marshaling and make actual calls on the objects; and (7) standard marshaler that marshals interface pointers into object references on the server side and marshals the object references on the client side. Note that interface proxies and stubs are application-specific and are generated by running an Interface Definition Language (IDL) compiler on application-supplied IDL files. The other objects are application-

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

independent and are provided by COM.



Purpose: COM is a platform-independent, distributed, object-oriented system for creating binary software components that can interact. COM is the foundation technology for Microsoft's OLE (compound documents) and ActiveX (Internet-enabled components) technologies.

Distributed components object model: Distributed Component Object Model (DCOM) is a proprietary Microsoft technology for communication among software components distributed across networked computers. DCOM, which originally was called "Network OLE", extends Microsoft's COM, and provides the communication substrate under Microsoft's COM+ application server infrastructure. It has been deprecated in favor of the Microsoft .NET Remoting, a part of their .NET Framework.^{[1][2]}

The addition of the "D" to COM was due to extensive use of DCE/RPC (Distributed Computing Environment/Remote Procedure Calls) – more specifically Microsoft's enhanced version, known as MSRPC.

In terms of the extensions it added to COM, DCOM had to solve the problems of

- **Marshaling – serializing and deserializing the arguments and return values of method calls "over the wire".**

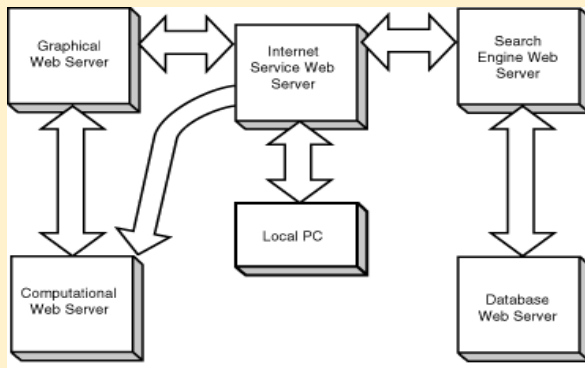
INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

- **Distributed garbage collection** – ensuring that references held by clients of interfaces are released when, for example, the client process crashed, or the network connection was lost.
- **Aggregating hundreds or potentially tens of thousands of references to objects held by clients of interfaces at a single host, into a single "ping" function, in order to minimize bandwidth utilization.**

One of the key factors in solving these problems is the use of DCE/RPC as the underlying RPC mechanism behind DCOM. DCE/RPC has strictly defined rules regarding marshalling and who is responsible for freeing memory.

DCOM was a major competitor to CORBA. Proponents of both of these technologies saw them as one day becoming the model for code and service-reuse over the Internet. However, the difficulties involved in getting either of these technologies to work over Internet firewalls, and on unknown and insecure machines, meant that normal HTTP requests in combination with web browsers won out over both of them. Microsoft, at one point, attempted and failed to head this off by adding an extra http transport to DCE/RPC called ncacn_http (Network Computing Architecture, Connection-based, over HTTP). This was later resurrected to support a Microsoft Exchange 2003 connection over HTTP.



Q15. Write a simple program to demonstrate the COM?

Ans: Option Explicit

Private mdbCelsius As Double

Private mdbFahrenheit As Double

Public Event BelowFreezing()

Public Event AboveBoiling()

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Public Property Get Celsius() As Double

Celsius = mdblCelsius

End Property

Public Property Let Celsius(NewTemperature As Double)

mdblCelsius = NewTemperature

mdblFahrenheit = ((NewTemperature * 9) / 5) + 32

If mdblCelsius < 0 Then

RaiseEvent BelowFreezing

End If

If mdblCelsius > 100 Then

RaiseEvent AboveBoiling

End If

End Property

Public Property Get Fahrenheit() As Double

Fahrenheit = mdblFahrenheit

End Property

Public Property Let Fahrenheit(NewTemperature As Double)

mdblFahrenheit = NewTemperature

mdblCelsius = ((NewTemperature - 32) * 5) / 9

If mdblFahrenheit < 32 Then

RaiseEvent BelowFreezing

End If

If mdblFahrenheit > 212 Then

RaiseEvent AboveBoiling

End If

End Property

Public Function GetCelsius() As Double

GetCelsius = mdblCelsius

End Function

Public Function GetFahrenheit() As Double

GetFahrenheit = mdblFahrenheit

End Function

Private Sub Class_Initialize()

mdblCelsius = 0

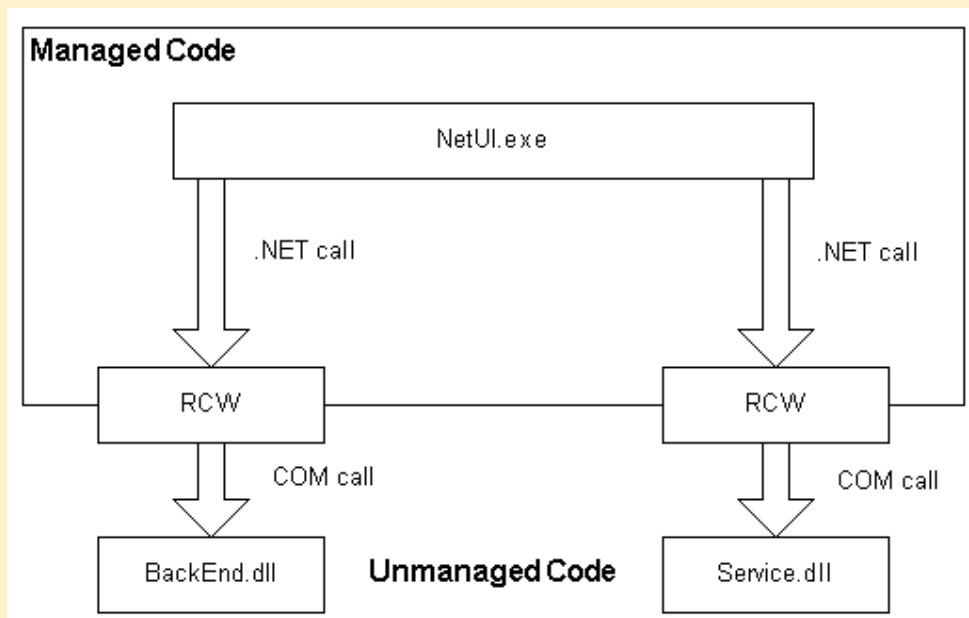
mdblFahrenheit = 32

End Sub

Manage and Unmanaged coding in Component object model:

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA



Q16. What is security in terms of COM/DCOM? How it can be maintained in COM/DCOM?

Ans: Security in COM is firmly based on the security provided by Windows and the underlying RPC security mechanisms. COM security relies on authentication (the process of verifying a caller's identity) and authorization (the process of determining whether a caller is authorized to do what it is asking to do). There are two main types of security in COM: activation security and call security. Activation security determines whether a client can launch a server at all. After a server has been launched, you can use call security to control access to a server's objects.

In this security model, servers manage and help protect objects, clients get access to objects through servers, and servers can attempt access while impersonating the client.

The system implements the Kerberos v5 authentication protocol and the Channel security package. It also includes features such as delegate-level impersonation, mutual authentication, the ability to set authentication levels for an AppID in the registry, and cloaking. Using COM security, you can implement objects that can perform privileged operations without compromising security.

Because there is a wide range of COM security features available, it is helpful to initially determine what kind of security your application needs. For most applications, setting an acceptable level of security can be a painless process, but you can also use COM security to support very complex security scenarios.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

How you set up COM security for your application depends on what kind of security your application needs. There are several common situations that determine what you should do.

If you decide to use the COM security defaults, you do not have to do anything—COM handles it all. For information on what these default settings are, see COM Security Defaults.

You can also prevent any remote calls into your machine by disabling DCOM altogether (COM between remote computers). For more information, see Setting System-Wide Security Using DCOMCNFG.

For legacy or new applications, you can set process-wide security in the registry. For more information, see Setting Process wide Security Through the Registry.

You can also override default security settings for calls to certain interfaces in the process while setting default security for the remainder of the process (to allow COM to handle the general cases). For more information, see Setting Security at the Interface Proxy Level.

For complex security requirements, you can handle all security programmatically rather than allowing COM to handle it for you. To do this, call Initialize Security to disable automatic authentication, and then control all the security settings by setting security on a per-interface proxy basis. For more information, see Setting Process wide Security with Initialize Security and Setting Security at the Interface Proxy Level.

In some scenarios, you might want to turn off security completely. You might decide that your application does not need any security, or you might want to disable security during development time so that you can enable security features individually. To learn how to disable COM security, see Turning Off Security.

Security in COM relies on authentication services administered by security packages. NTLMSSP works well for many applications but does not provide the more robust security offered by other packages. Therefore, COM supports the Schannel security package and the Kerberos v5 security protocol. For more details on using these security packages, see COM and Security Packages.

You can use the COM security defaults for your application rather than specifying your own security settings. In that case, COM will initialize and manage security for you. You do not need to configure the registry or call any security functions in your program.

However, if certain registry named values have been set or modified, the security defaults that COM uses will be affected. The list below describes

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

COM security default values and explains how some values are influenced by registry settings.

Following are the default security values that COM uses:

- **The default security service provider is the one that is determined by COM to be the most compatible with the environment. COM chooses either the Kerberos v5 protocol or NTLMSSP, with the Kerberos protocol being the default choice. None of the protocols provided by Channel are ever chosen as the default.**
- **The system identifies a caller through user name and password and automatically creates an identification token used by the security system.**
- **If the Legacy Authentication Level named value exists and if its value has been set, that value is used. Otherwise, the authentication level is set at connect (RPC_C_AUTHN_LEVEL_CONNECT). This level means that at the first call a client makes to the server, COM does an authentication check. If the client passes the check, no further authentication is done. The Authentication Level value can also be set under the App ID key.**
- **If the Legacy Impersonation Level named value exists and if its value has been set, that value is used. Otherwise, the impersonation level is set to identify (RPC_C_IMP_LEVEL_IDENTIFY). Impersonation rights are granted by the client to the server. Identify level means that the server can obtain the client's identity. The server can impersonate the client for access control list (ACL) checking but cannot access system objects as the client. For more information, see Impersonation Levels and Cloaking.**
- **If the Access Permission named value under App ID exists and has been set, that value is used. Otherwise, COM checks for a Default Access Permission entry. If present, that value is used. If this value is not present, COM constructs an ACL that grants permissions to the server identity and the local system.**
- **If the SRPTrustLevel named value under AppID exists and has been set, that value is used. Otherwise, the Software Restriction Policy (SRP) trust level is set to Disallowed (SAFER_LEVELID_DISALLOWED), which indicates that the application is run in a constrained environment and is disallowed from accessing any security-sensitive user privileges of the user.**

Dcomcnfg.exe provides a user interface for modifying certain settings in the registry. By using Dcomcnfg.exe, you can enable security either on a computer-wide or a process-wide basis. You can enable security for a particular computer so that when a process does not provide its own security settings, either programmatically or through registry values, the

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

values set by Dcomcnfg.exe will be used. Or you can use Dcomcnfg.exe to enable security for a particular application only.

When enabling security, there are two primary tasks to accomplish:

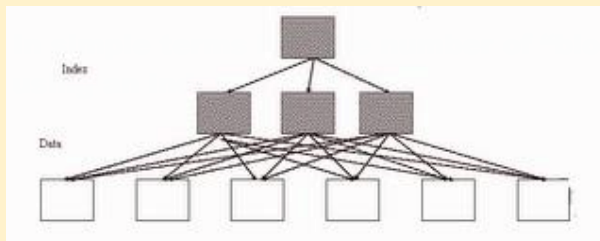
- Set an authentication level that is not None.
- Set permissions, including both launch and access permissions.

The steps taken to accomplish these tasks depend on whether you are enabling security for the whole computer or just for a particular application. Also, you may want to set other values for the computer or application.

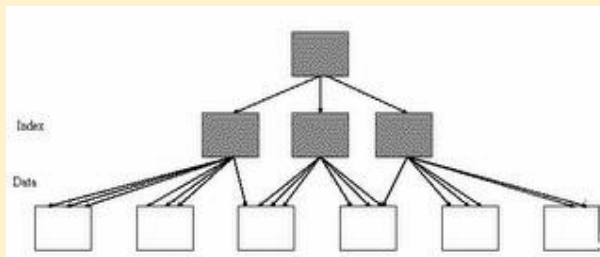
Note You must be an administrator to run Dcomcnfg.exe.

Q17. What is clustering factor? How is it beneficial in COM programming?

Ans: : The clustering factor is a number which represent the degree to which data is randomly distributed in a table.



“block switches” while reading a table



scatter the rows of the table are. points to the first data block and ta block. So while making index have to switch between blocks more than once because rows are scatter. So the number of times optimizer will make these switches is actually termed as “Clustering factor”.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Figure: Good clustering factor

The above image represents "Good CF". In an event of index range scan, optimizer will not have to jump to next data block as most of the index entries points to same data block. This helps significantly in reducing the cost of your SELECT statements.

Clustering factor is stored in data dictionary and can be viewed from dba_indexes (or user_indexes)

```
SQL> create table sac as select * from all_objects;
```

Table created.

```
SQL> create index obj_id_indx on sac(object_id);
```

Index created.

```
SQL> select clustering_factor from user_indexes where  
index_name='OBJ_ID_INDX';
```

CLUSTERING_FACTOR

545

```
SQL> select count(*) from sac;
```

COUNT(*)

38956

```
SQL> select blocks from user_segments where
```

```
segment_name='OBJ_ID_INDX';
```

BLOCKS

96

The above example shows that index has to jump 545 times to give you the full data had you performed full table scan using the index.

Note:

- A good CF is equal (or near) to the values of number of blocks of table.
- A bad CF is equal (or near) to the number of rows of table.

Myth:

- Rebuilding of index can improve the CF.

Then how to improve the CF?

- To improve the CF, it's the table that must be rebuilt (and reordered).
- If table has multiple indexes, careful consideration needs to be given by which index to order table.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Clustering is a technique used to make several computers act as one larger machine, splitting tasks amongst them. They allow one to take several cheap stations, and combine them together to a larger system. It also allows for more redundancy for the system - if one machine in the cluster dies off, the other machines can cover up for it until the malfunctioning machine is repaired, and all this without bringing the whole system down. This type of setup is thus common in systems that must run 24 hours non-stop.

Clustering is often implemented in software, often using a protocol named PVM to communicate between the different machines. Examples for such systems are Beowulf, for Linux systems, or the clustering system by Tandem corporation.

Q18. What is Active X control and Active X DLL? Describe?

Ans: A Microsoft ActiveX control is essentially a simple OLE object that supports the un known interface. This section offers solutions for making a control work well in the Internet environment, with the ultimate goal of delivering optimal quality of service to users. For example, because browser speed is one of the primary factors in users' perception of quality, this section aims to provide solutions that allow an HTML document or page to become visible as soon as possible and interactive very shortly thereafter, while allowing controls to retrieve large data blocks in the background.

About ActiveX Controls

This section introduces ActiveX Controls and provides general guidelines for control developers. It includes information about run-time licensing for an ActiveX control, identifying an ActiveX control as safe for scripting or safe for code download, packaging an ActiveX control in a .cab file, and applying a test certificate to an ActiveX control.

About Control Instantiation

How does a container bring control class code (binary executable code) into memory and instantiate a control of that class?

About Cooperative and Asynchronous Data Retrieval

To deliver optimal quality of service to browser users, a control must read data from an external source in a cooperative manner as much as

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

possible. That is, the control should support reading linked data in an asynchronous manner, paying attention to container prioritization and allowing the container to participate in the transfer as it wants. To provide the container with authority over the binding operation, the control should use

the `IBindHost::MonikerBindToStorage` and `IBindHost::MonikerBindToObject` methods to bind to any moniker.

About Internet COM Object Requirements

There is no question that OLE Controls, or Component Object Model (COM) objects in general, are useful as part of a document or page on an Internet site. However, some problems exist with regard to how a control is specified within a document, and how a control can behave well in a slow-link environment such as the Internet, including how the control retrieves its data in an incremental or progressive fashion, working well with other controls that might also be retrieving their data in the same manner.

About IObject Safety Extensions for Internet Explorer

This document describes the extensions to the `IObjectSafety` interface that support the new security features implemented in Microsoft Internet Explorer 4.0.

Added Standard Properties, Methods, Events, and Interfaces

The following table summarizes the added standard dispIDs described in this article.

Building ActiveX Controls for Internet Explorer

This article covers features of Windows Internet Explorer that a developer writing ActiveX Controls should take into account when targeting Internet Explorer as a container. Some of the material in this article discusses existing technologies that have increased relevance with Internet Explorer; other sections discuss new technologies and services that are available only to a control in Internet Explorer.

Communicating Control Readiness

For all persistence interfaces other than `IPersistMoniker`, the authoring tool or container assumes that once `IPersist*::Load` returns, the control has loaded all its properties. However, controls that use data paths might not actually have all of their data at this time.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Component Categories for Describing Internet-Aware Objects

A category identifier (CATID) is a GUID that is stored in the registry as an attribute of a class. There are various useful categories for introducing a ActiveX control to its container; they describe which features the object requires and those it implements.

Data Path Properties

This article is, however, concerned with the more general problem of retrieving any significant amounts of data in a cooperative fashion from possibly many distributed locations after the control has already been instantiated and is possibly interacting with the container and the user in other ways. This capability has been called "Progressive Rendering," "Progressive Property Disclosure," and "Progressive Downloading." The term "Progressive Data Retrieval" will be used to refer to all of these more special cases at once, because it makes no assumptions about the exact type of data that is being retrieved—for example, it might be properties, might be images, might be anything.

Designing Secure ActiveX Controls

Any ActiveX control should be conceived and designed with security in mind.

How to Register an ActiveX Object as the Player for a Media Type

This topic describes how to register a ActiveX object as the viewer or player for a particular media type (MIME). This registration is essential for Internet Explorer to launch the correct player when interpreting the standard HTML A HREF tag or the Netscape-introduced EMBED tag. Without this registration, it is impossible for Internet Explorer to guess what application to use to display or play an unknown media file.

Internet Awareness for Objects, Controls, and Containers

The following documents, which are referenced in text using shortened names, contain information relevant to COM objects.

Introduction to ActiveX Control Containers

This article is an overview for ActiveX control containers. The following sections explain how to implement and program ActiveX control containers that will function well with other controls and containers.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Introduction to ActiveX Controls

The purpose of this topic is to provide guidelines for implementing ActiveX Controls that interoperate well with containers and other controls. This article defines the minimum set of interfaces, methods, and features that are required of ActiveX Controls to accomplish seamless and useful interoperability.

Licensing ActiveX Controls

This article explains the licensing strategy for ActiveX controls. In addition, it describes the functionality that must be implemented on an ActiveX control in order to support this strategy.

Managing ActiveX Control Loading

Like other software programs, ActiveX controls may contain vulnerabilities. Loading vulnerable ActiveX controls into Internet Explorer can place your computer at risk. This article shows how to prevent Internet Explorer from loading vulnerable ActiveX controls. It also shows how to redirect such requests to load alternate, more secure ActiveX controls.

Non-Admin ActiveX Controls

For many existing and future ActiveX controls, Internet Explorer 8 removes administrative involvement from the installation process. The solution addresses issues raised by enterprise customers who used previous versions of Internet Explorer, where administration of controls was not optimal. Non-administrative ActiveX controls (or per-user ActiveX installation) are available on Windows Vista and later.

Packaging ActiveX Controls

This article introduces a data-compression technology and associated tool set that you can use to package your ActiveX control for faster, more efficient downloading over the Internet or an intranet.

Palette Management for ActiveX Objects

This document describes palette management for ActiveX objects, specifically ActiveX Controls and Active Document objects. The palette management rules described herein allow multiple controls inside a form or an HTML page to display correctly, while still leaving enough flexibility

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

to allow a control to demand palette control. In general, containers are responsible for palette management, and contained objects (for example, controls) should only realize their palettes in the background.

Per-Site ActiveX Controls

Internet Explorer 8 allows greater control of where and under what context ActiveX controls can run. In this version of Internet Explorer, ActiveX controls that are embedded as Web objects are presented to the user as add-ons. Through the new Manage Add-ons dialog box, the registry, or site-locking technology (such as the Site Lock ATL Template), these add-ons can be restricted for use on specific Web sites.

Safe Initialization and Scripting for ActiveX Controls

This article describes the code that a control developer should implement to ensure safe initialization and safe scripting for a ActiveX control.

Security Considerations: ActiveX: This document provides information about security considerations related to ActiveX.

Standard Internet-Aware Objects: As a benefit to control developers, Microsoft will supply Internet-aware picture, sound, and video objects that will make dealing with these data formats in the Internet environment a non-issue for many controls. The existing picture object, for example, makes a control's manipulation of a bitmap, metafile, or icon nearly effortless. The additional objects described here aim to do the same for their respective data types, where all standard objects will support the ability to retrieve large data BLOBs in an asynchronous manner.

Data: When a control has any persistent data at all, the class identifier (CLSID) element always exists along with one or both of the other two elements. How these elements are stored in relation to the container document involves the concepts of embedding and linking, which originated in OLE Documents (the OLE compound document architecture), described in a following section.

Supporting Offline Browsing in Applications and Components: This article describes the API elements that enable third-party developers to hook into the offline browsing architecture. It also provides some recommendations to help developers provide a seamless experience for users.

INTERNAL ASSESSMENT & TERM WORK (5BCA1) SOLUTION

MAHARAJA COLLEGE ARA

Pages: This article describes how a developer can add ActiveX support to an existing OLE Automation control.

Active X DLL:

An ActiveX Dynamic Link Library (DLL) is part of a programming framework developed by Microsoft Inc., the company behind such programs as Windows. DLLs are small, self-contained programs designed to allow communication between computer applications, either over the Internet or corporate intranet, usually via a web browser. The most common use for ActiveX DLLs is to allow the embedding of interactive functions, such as video players, into web browsers.