**Documentum Interview Questions**

**Q. What is Documentum?**  
Documentum is enterprise content management software that provides management of content and attributes such as check-in, check-out, workflow and version management. Documentum provides management capabilities for all types of content including business documents, images, video, e-mail, Web pages, fixed content, XML-tagged documents, etc. The core of Documentum is a repository in which the content is stored securely under compliance rules.

Documentum also provides a suite of services which include document management, collaboration, search, content classification, input management, business process management (BPM), customer communication management, Web content management, digital asset management, forms processing, information rights management, compliance and archiving etc.

**Q. What is a content server?**  
Content Server is the foundation of Documentum’s content management system. It is a multi-featured product that provides access to the data and content files that constitute a DOCBASE. It stores content files, their indexes and properties in the docbase. So it controls the access to the docbase. It controls the user’s access to objects in the repository by providing ACL. It is the core functionality that allows users to create, capture, manage, deliver and archive enterprise content.

**Q. What is a docbase?**  
The place where the information about a document is stored is called as Docbase. It’s the combination of the operating system (file system) and the database. The actual file (content) is stored in the file system and the information or attributes of the document are stored in Data base.

**Q. What is DFC?**  
DFC – Documentum Foundation Classes.  
This is an object-oriented application programming interface (API) and framework for accessing, customizing and extending Documentum functionalities, implemented as a set of Java interfaces and implementation classes.

**Q. What do you understand by WDK?**  
WDK – Web Development Kit  
It is a developer’s toolkit that facilitates the development of complex web based applications that connect to Documentum content repositories and Documentum content server. It contains large library of reusable components and controls to perform common Documentum management functionalities.

**Q. What is a docbroker/Connection broker?**  
Connection broker primarily know as docbroker provides repository connection information to the client application. Before installing a WDK application, one needs to obtain the names of the connection brokers to which the appropriate repositories project and the port numbers on which the connection brokers listen. The connection brokers used by WDK applications are defined in the dmcl.ini file on the application server host.

**Q. Explain how content server and docbase communicate with the database?**  
Documentum Content Server requires a properly configured database. The database tables store attribute values for each object in the repository.

Documentum provides a single repository for content and metadata. It uses an extensible object-oriented model to store content and metadata in the repository. Everything in a repository is stored as objects. To retrieve metadata, Document Query Language (DQL) is used. DQL is a superset of SQL that provides a single, unified query language for all the objects managed by Content Server. The repository owner (database user) account is the account used by Content Server to connect to the RDBMS.

**Q. What is a DocApp?**  
A DocApp is nothing but a packaging unit for Documentum objects. DocApps are used to package various elements for customization (such as custom Object Types, Lifecycles, etc.) so that they can be ported from one repository to another easily. Using Documentum Application Builder (DAB) you can create and package all your objects within a DocApp. A DocApp archive is a portable representation of a DocApp in a file on the file system. DocApps simplify developing and redeploying applications.

**Q. What is the significance of connection pooling in Documentum?**  
Connection pooling is an optional feature of Content Server that allows a primary repository session to be recycled and used by more than one user. Implementing connection pooling provides performance benefits for applications that execute frequent connections and disconnections for multiple users. When connection pooling is enabled in the DMCL, primary repository sessions are not closed when a user disconnects. Instead, they are held in a connection pool created by the DMCL. When another user requests a connection to the repository, the user is authenticated and given the free connection. When a user disconnects or a new user assumes ownership of a primary repository session, all sub-connections open in that session is closed.

**Q. Do you know how Documentum manages the passwords?**  
Documentum supports a variety of options for implementing user authentication:  
1) In-line password: A user can be authenticated using an encrypted password that is stored in the user\_password attribute of the user object.  
2) LDAP directory server:  If you use LDAP directory server, you have the following options:  
Authenticate against the directory server directly, using a secure or a non-secure connection  
Authenticate using an LDAP-enabled dm\_check\_password program  
3) Custom password checking program: One can create a custom password checking program and set up the servers to call that program for user authentication. This option is particularly useful if you want to use Windows domain authentication for UNIX users.  
4) Authentication plug-in: Documentum provides authentication plug-in which supports Single Sign-On and strong authentication. (Strong authentication is the use of authentication tokens such as smart cards or biometrics.)

**Q. How can applications use Documentum?**  
Custom applications can be developed using DFC (Documentum Foundation Classes) or WDK (Web Development Kit). DFC exposes Content Server API while WDK offers a framework and model for building a web application.

However, there are numerous aspects and options when developing a custom application using Documentum. Custom workflows can model business processes and lifecycles can encapsulate business rules around documents. SBO’s (Service-based Business Objects) and TBO’s (Type-based Business Objects) can encapsulate business logic.

Documentum capabilities can also be encapsulated in Web Services opening up a vast array of possibilities. D6 (Documentum 6.0) offers DFS (Documentum Foundation Services) which includes some core services as well as tools for developing web services for Documentum. Then there are standard interfaces such as FTP and JDBC, which allow a Documentum repository to be accessed using these protocols. Specialized connectors and portlets offer additional alternatives. PIA (Primary Interlope Assembly) allows access from .NET platform

**Q. What’s the difference between a component and a container?**  
A component is composed of one or more JSP pages, supporting behaviour classes and an XML configuration file.

A container is a specialized component. Many components share common UI and state. For example, dialogs have a title, content area, and a button panel. Containers provide these common layout and state for multiple components.

Components can be used within more than one container, inheriting their UI and state from the container.

**Q. What are a component’s main features?**  
A component consists of  
1) A component definition within a XML file that defines the following elements:  
Unique . The optional component inheritance from a parent component definition is specified in the “extends” attribute.  
element which defines the context in which the component definition is applied.  
element containing the parameter used in the behaviour class.  
element contains all presentation pages.  
element specifying the behaviour class  
element specifies the class that contains externalized strings for the component class and JSP pages. Properties files in the bundle can be localized.  
2) JSP pages: Layout is defined in the JSP pages, using HTML and configurable Documentum tags. The definition defined in the element of the    definition XML as stated above.  
3) Component behaviour class: has the logic of the component. This is defined in the element of the XML file as stated above.  
4) String resource file defined in the element of the XML configuration

**Q. What are the major lifecycle methods of a component in Documentum?**  
The major lifecycle methods of a component are listed below:  
OnInit, onRender, onRenderEnd, onCommitChanges, onOk, onCancel etc.

**Q. What do you mean by Documentum object model?**  
The Documentum object model is used to represent content in the repository. The object model is hierarchical as in an object-oriented programming language. Child objects inherit all attributes and behaviors of the parent object.

**Q. Describe the Documentum security model? OR How many level of ACL does exist?**  
Access Control lists (ACL) provides the security of Documentum objects. Every sysobject is assigned an ACL which has information about the users and groups access on that object. There are basic 7 different levels of permissions which can be assigned to a user or a group of users for an object. The following are the object level permissions:

|  |  |  |
| --- | --- | --- |
| Level | Permission | Description |
| 1 | None | No access is permitted. |
| 2 | Browse | The user can look at attribute values but not at associated content |
| 3 | Read | The user can read content but not update. |
| 4 | Relate | The user can attach an annotation to the object. |
| 5 | Version | The user can version the object. |
| 6 | Write | The user can write and update the object. |
| 7 | Delete | The user can delete the object. |

Other than these basic object level permissions we have extended permission levels also in Documentum. The extended permissions are described below:  
execute\_proc    : Allows the user to execute the procedure (if it is a procedure)  
change\_location: Allows the user to change the location of the document  
change\_state    : Allows the user to change the state of the document using the document lifecycle  
change\_permit  : Allows the user to change the object’s permissions  
change\_owner  : Allows the user to change the owner of the object

**Q. List few native DCTM types you have commonly used?**  
The main native Object types in Documentum are:  
1. dm\_sysobject  
2. dm\_folder  
3. dm\_cabinet  
4. dm\_document  
5. dm\_user  
6. dm\_audittrail

**Q. What tools can be used to create custom object types?**  
DAB – Documentum Application Builder  
DQL scripts

[**WDK**](https://mindmajix.com/documentum)

**Q. Do you need a container to write a component?**  
No, a container is not mandatory when writing a component. E.g. in Webtop the “Copy to Clipboard” feature doesn’t have a container.

**Q. What is a precondition and what is its importance? When is a precondition executed?**  
A precondition can be written for an action which determines whether the context is valid for the control or not. Accordingly it will then render an action control as enabled or disabled. They are however optional in action definition. If not mentioned the action will always execute.  
Preconditions are called for each item in a list of component. If there are 10 items and 20 applicable actions, 200 preconditions will be executed before the list is rendered. So it can affect application performance if not used correctly.

**Q. Describe the version management capabilities of Documentum?**  
Versioning is an important content management feature of Documentum which helps in managing and tracking multiple versions of a document. The check-in / check-out option provides versioning capability in Documentum. The version label attribute is the heart of the version management system which is used by the content server to label multiple versions of the document. Each label can have an implicit label or a symbolic label. The implicit label is assigned by the server whereas the symbolic label is assigned by the user.

**Q. What is the significance of i\_chronicle\_id?**  
The i\_chronicle\_id attribute value is same for all the versions of a document. Each versioned object is assigned a unique “r\_object\_id” value but “i\_chronicle\_id” is the one attribute which is same for all versions of a document and binds all versioned objects together.

**Q. What are implicit and symbolic labels in Documentum version management?**  
Implicit Labels: The implicit version label is a numeric label. It is generally assigned by the server and is always stored in the first position of the r\_version\_label attribute. By default, the first time you save an object, the server sets the implicit version label to 1.0. Each time you check out the object and check it back in, the server creates a new version of the object and increments the implicit versions label (1.1, 1.2, 1.3, and so forth).  
Symbolic Labels: A symbolic version label is either system or user-defined. Using symbolic version labels, you can provide labels that are meaningful to your application and work environment. Symbolic labels are stored starting in the second position (r\_version\_label [1]) in the r\_version\_label attribute.

**Q. What are renditions?**  
A rendition is a representation of a document that differs from the original document only in its format or some aspect of the format. It is an alternate copy of a file or an additional file that can be included with an object. For example, it may be a copy of an image in a different format or in a different resolution.

**Q. What is app.xml and web.xml?**  
App.xml: The app.xml is an application configuration file wherein one can configure the application wide behaviour. In Documentum each application layer (webtop, wdk, webcomponent etc…)  has an app.xml file. It is made of application elements tag such as for failover, content\_transfer, authentications, formats, plug-in, listeners etc. To inherit and override settings in another application layer, the application definition can extend an application definition in another layer.  
Web.xml: The web.xml file is a Web Application Deployment Descriptor for a web application. The web.xml file provides configuration and deployment information for the Web components that comprise a Web application. Examples of Web components are servlet parameters, servlet and Java Server Pages (JSP) definitions, and Uniform Resource Locators (URL) mappings.

**Q. What is a workflow and how can we create workflows in Documentum?**  
A workflow is a process that electronically passes documents and instructions from user to user. A workflow automates the process, ensuring that the right file goes to the right people in the right order.  
Workflow tools:  used to create the workflow definition called the workflow template are:  
1)    Workflow Manager  
2)     Business Process Manager

**Q. What is a lifecycle and how can we create lifecycles?**  
A document lifecycle is a sequence of states a file goes through between its creation and expiration. When you create a file, the system assigns a document lifecycle to the file and puts the file into the first state in the lifecycle. Typical lifecycle states include WIP (Work In Progress), indicating a document is in its draft phase, and Staging, indicating a document is complete and ready for approvals.

**Q. What is the difference between a workflow and lifecycle?**  
Lifecycles and workflows are tools for enforcing business rules within Documentum.  
A lifecycle is a sequence of a file going between its creation and expiration states. Typical lifecycle states include Draft, Work in Progress, Approved, released etc. Thus the focus of the lifecycle is an object.  
A workflow represents a network of activities that electronically passes documents and instructions from user to user. It automates the process ensuring the correct flow of work in a system. For example, an approval workflow can take a document from the author who created the document to an editor for review and lastly a manager for approval. Thus the workflow process may result in changing lifecycle states of the document.

**Q. Describe how you would use and leverage third-party data in Documentum?**  
The third party data from an external table can be accessed from Documentum docbase with the help of Register tables. One can register any external table in Documentum and can thereby access the data residing in it as simply as it is residing in a Documentum docbase. To register an external table in Documentum we use the following DQL:   
DQL>    “Register table dm\_dbo.table\_name (columname datatype)”  
The above query registers the table in documentum database and as known this will create an object of type “dm\_registered”.

**Q. Differentiate between role and group?**  
A group in Documentum is a convenient way of collecting all users with the same permission level. A group can thereby facilitate managing permissions (assigning ACL’s), assigning a task in a group work basket in a workflow, as an owner allowing all members of a group to have ownership for the document. Thus groups are mainly used for Object level permissions.  
Roles are mainly used for providing function permissions. They are basically a special type of groups. It is created by setting the group\_class attribute to role and the group\_name attribute to the role name.

**Q. What do you know about UCF?**  
UCF – Unified Content Facilities  
It is a lightweight applet that is responsible for transferring content between the content server, application server and the client machine during operations such as check-in, checkout, import etc. UCF is a Java-based client application that is installed at run-time. Since WDK application uses UCF content transfer it downloads a lightweight applet to the browser when the client uses the Documentum operations for the first time.

**Q. What are virtual documents?**  
A virtual document is a document that contains components or children documents. It is composed of other components, each component being an individual object. The dmr\_containtment object stores information about each individual component of a virtual document.  
Virtual documents are used mainly for the following scenarios:  
1) When there are multiple authors of a document, virtual document creates individual component providing owner, security and other characteristics for the document.  
2) Virtual documents are useful in managing multiple file formats For E.g. if there are sub attachments like an excel sheet and a presentation attached to some word document. In this case a virtual document creates three heterogeneous file formats keeping a single logical document.  
3)  Virtual documents are useful when order of the children matters. By default the ordering depends on adding or removing sequence of the components. But it can also be done manually.

**Q. What are the major difference in Documentum 5.3 and Documentum 6.x?**  
Following are some of the major features introduced in Documentum 6.x which were not present in 5.3  
– Webtop performance improved by reducing the number of refreshes and by making better utilization of the AJAX framework  
– Documentum functionality presented as service in 6.x (Documentum Foundation Services i.e. DFS)  
– Introducing Documentum Composer – which Supplies an Eclipse-based platform for Documentum tools, allowing you to configure application elements such as workflow templates, lifecycle definitions, security settings, aspects, and object types  
– Introduces Aspects – new member in BOF family. Aspects are a mechanism for adding behavior and/or attributes to a Documentum object instance (and not object type) without changing its type definition  
– XML Store as the underlying architecture for metadata and Web 2.0 content types in D6  
– In the Documentum 6.5 release, the DocApp was migrated to the new Documentum Archive (DAR) format, which is compatible with Eclipse?based Documentum Composer development tool  
– Unification of dfc.properties and dmcl.ini file. The dfc.properties file replaces dmcl.ini file to define behavioural preferences for client and application  
– New DFC tracing capabilities and configuration. Tracing is now controlled by entries in dfc.properties rather than log4j file.

**Documentum**

Today, in any organization, content management has become a tedious task as a lot of content is being produced daily. It be either managing some content or making its lifecycle a fruitful pitch for the company, the IT professionals have to engage a lot of their time in doing it the right way. In order to overcome the challenges faced with traditional content management techniques, Documentum has been introduced over the time.

The organizational content can be emerged from different sources such as CRMs (Customer Relationship Management) or ERPs (Enterprise resource planning). Hence, we can simply term Documentum as a panacea to manage every kind of digital content used in the organization. Using Documentum also helps in producing a more efficient decision-making process.

Documentum works as an Enterprise Content Management (ECM) programme, which is used to store valuable content from all over the enterprise in a safe and secure place called repository. Along with the storage units, it also provides ease-of-use for controlling and organizing files. You can easily edit, recover, utilize or retrieve any important information from anywhere and anytime you desire. Furthermore, it also helps in making your document secured from external and internal risk factors. Here, in this blog, we are going to have a brief look at features, architecture, advantages, and disadvantages of this ECM platform termed as Documentum.

**Table of Contents**

* **What is meant by Documentum?**
* **Advantages of Documentum**
* **Documentum Architecture**
  + **Client Application**
  + **Documentum server**
  + **RDBMS and File System**
* **Layers of Documentum**
  + **Services Layer**
  + **Interface Layer**
  + **Client Layer**
  + **Application Layer**
* **Documentum Foundation Classes**
* **Life Cycles of Documentum**
* **What is meant by Web Services?**
  + **Documentum Web-Based Applications**
  + **Documentum Microsoft Windows-Based Applications**
  + **Important Products of Documentum**
  + **Documentum WebTop**
  + **Documentum Web Publisher**
  + **Documentum Digital Asset Management (DAM)**
* **Features of Documentum**
* **Benefits of Documentum**

What is meant by Documentum?

Documentum system was developed by John Newton and Howard Shao in 1990, in order to solve the unorganized content problem using relational database technologies. Originally, this system was developed for a company ‘Boeing’ (an aerospace Company) which has a large amount of data in the form of paper documents, information, and training manual. Its main function was to help the company in storing, maintaining, and organizing thousands of pages of information into a common repository.

The Documentum software also helps in converting paper-based documents into application-ready files allowing a user to save a lot of time, cost and effort.  Enterprise’s ECM system is used to manage the content from other applications from across the organization such as from ERP’s, CRM systems and portals. Furthermore, documentum provides insight into the organization, where you can easily view, edit or update any document shared with you through the Documentum system.

For Example, if you are using Documentum system to edit a file you can “**check it out**” provided no other member of the company is editing the file. It also blocks others from editing the same file if you are editing it. Moreover, you are also allowed to check out the previously made edits on the document and by whom those edits were made. For us, the users, it looks like the system stores only one single file.

However, Documentum does have all the previously edited versions of the file. You can also add a small comment mentioning what edits you have made, to differ the current versions from the previous ones. The current version of the ECM in the market is known as Documentum D2 16.4 released in 2018.

Advantages of Documentum

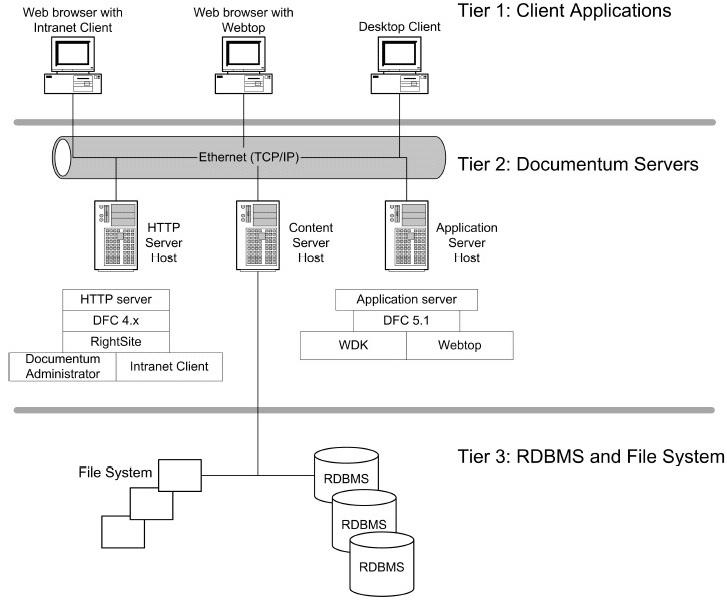
Here are a few advantages of using Documentum instead of any other Content Management Software.

* **Flexible and Supportive Content Capabilities:** Documentum’s most advanced ability includes workflow features, lifecycle, and auditing which can be used as per your requirements.
* **The Spreadsheets are controlled and organized:** Its system is configured in such a way that it makes the creation and control of highly complex and important documents easier and simpler than any other CMS.
* **Personalized and User-Friendly:** As it has the capability to adapt according to the requirement, it makes the interface very much user-friendly.
* Highly Configurable Documentation: It also has the feature of configuring documents according to specifications provided.
* **Advanced Usability for Virtual Documents:** This system helps in simplifying the use of virtual documents (i.e. a document containing parts of other documents from different authors), throughout the organization.
* Enhanced User Interface: Widgets and easy-to-use tools help in navigating the software faster.
* **Automated Features:** Numbering system or naming a file are made automated which helps in reducing time and efforts.
* **Security:** EMC enforces a high level of security as these files are kept safe in the software’s controlled repository and also organized.
* **Smarter Display of Content:** Documentum provides a smarter way of viewing content. It previews the content in thumbnail form, that helps in a search operation for any document.
* **Search Engine:** EMC also has a search engine similar to the web search engine i.e. user can look for the file by looking for a certain keyword or phrase.
* **Workflow:** Other than routing of documents within the Documentum system, you can also communicate through Inbox. You can look at all the files you want and can approve or reject any file in the system.
* **Event Notification:** EMC’s system also provides a real-time update notification which provides you with a notification for all the changes made in the documents and by whom these changes are made by.

Documentum Architecture

The architecture of the Documentum consists of three basic stages namely:

* The Client application
* Documentum server
* RDBMS and File Systems



Client Application

Documentum system architecture generally follows the architecture of a client/server model. This is the layer where the client performs the operations such as edit, create and look up the document.

Here in this layer, we can interact with the server layer i.e. the content server in order to get our task done. The client-side sends a request to carry out the task such as editing a file on the server or retrieving a file and the server gets the task done.

Documentum server

Documentum Content Server is a new kind of content server that was designed to simplify the task of creating or editing by using any of the documentum application. This server handles the read and write tasks for web-based client applications.

The server uses HTTP or HTTPS protocol for all communications. The logic’s from application layer is transferred to the server and then calls are made using server API’s (DFC – Documentum foundation classes) to the particular repositories.

RDBMS and File System

RDBMS (Relational database management system) is a software developed for storing a large amount of data efficiently. The application layer performs various operations on the RDBMS. These operations can be easily accessed programmatically by an individual.

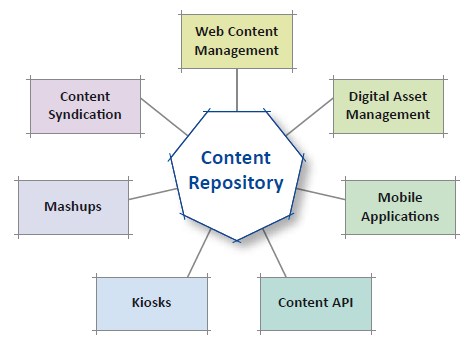
Situated at the bottom layer of architecture, the RDBMS and Operating system combinedly handle every request provided by the content server layer. It then stores valuable information, in the form of tables, in a relational database.

Layers of Documentum

There are four layers of Documentum Platform. Let's have a look at these layers below:

Services Layer

The Documentum content server is staged at the base of the Documentum platform which helps in managing the repository and implementing the core content capabilities. The content server then helps in availing these capabilities to the user with the help of the interface layer.



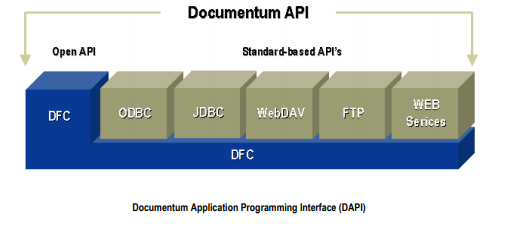
The Content repository uses a flexible model to store data and the associated metadata. The documents are made of content files having attributes (metadata). These attributes are collected as metadata which describes the content and the relationships between the content and other contents stored in the repository.

These metadata are then used by the user to search and retrieve the content relevant to their uses. The content stored in the particular repositories can be encrypted by the user in order to protect the data from any uncontrolled access.

Interface Layer

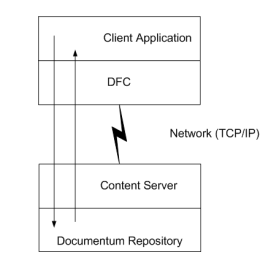
The interface layer is used by the web server clients to create a link and communicate with the content server and the content repositories. It is also known as the data link layer. This layer determines how a data bundles are packed and transmitted through the network. IEEE 802.2 and X.25 are some of the common link layer protocol on which the layer commonly operates.

The interface layer consists of the Documentum Foundation Classes (DFC) and many standard API’s (application programming interface),  which are used to control and access all of the functionality of the Documentum system. These API’s collective known as Documentum application programming interface (DAPI), which can be accessed even from the host of applications.



Client Layer

Client layer as explained before consists of the basic application and software for client usage. These applications are used to access Documentum repository. It includes end-user interfaces, administrator tools, and integrations with popular authoring tools and enterprise applications such as Documentum desktop or WebTop.



The components of the client layer are built on content management services available from the lower layers of Documentum architecture. The components in applications, that are built, can be common for other applications as both the web-based applications and Microsoft Windows-based applications access the same dictionary from the repository.

Application Layer

Documentum’s application layer provides the integration of content servers with enterprise applications. It also offers several products that provide a collaboration of content management system with the application layer. Here are some products:

* Documentum Content Services for SAP
* Documentum Content Services for Siebel e-Business Application
* Documentum Content Services for Lotus Notes Mail
* Documentum Content Services for Portals

The link is provided with the help of content services which gives the power of all the Documentum functionality to the user.

For example, a customer care executive using SAP or CRM application can check the customer's details, like the bill amount, the customer's segment, usage, and personal details including the phone number, address, instantly. Other than the executive, their project manager can review the operations and procedures with a single click on the report.

Documentum Foundation Classes

Documentum foundation classes, as mentioned in the interface layer, form a higher order of API providing all the functionality of  Documentum. DFC can be defined as a cluster of Java classes that can make the functionality of Documentum server available to you by implementing a set of attributes.

**The DFC can be used in following ways:**

* Accessing the functionality of Documentum.
* Customize the products of Documentum such as WebTop.
* You can execute the operation by writing a method or procedure to the content server as a part of the life cycle.
* Integration of Documentum application with third-party applications.

As DFC is based on Java, and it can easily integrate with the client programs that function on Java. In order, to run DFC API you must have a Java virtual machine (JVM) program on your system. The DFC can either be a machine that runs the content server or it can be a middle tier system i.e. application developed by the help of web development kit (WDK).

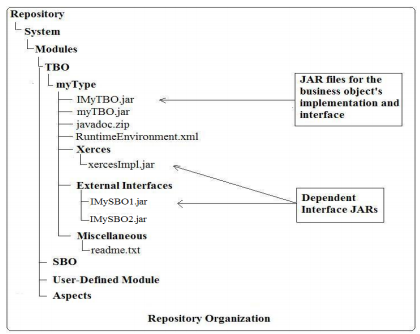
Documentum Business Objects

Documentum Business Object framework (BOF) is an object-oriented framework which can be built into DFC in order to develop a custom application. BOF in DFC have the capabilities of:

* Extending and enhancing the core functionality of Documentum
* Modify or add the types and behaviors of the existing Documentum.
* It also allows the client to add new attributes dynamically.

BOF provides the developer with the option of installing the BOF modules dynamically and directly on the storage. It allows you to use the application without adding any custom JAR files manually into each application. The dynamic usability of BOF module also provides you with the ability to update any module directly into the repository without shutting down the application.

The reusability factor of BOF module gives you access to its modules in any type of DFC based Documentum application. i.e. you don’t have to create a new module for every application. This feature reduces the time and effort and helps the user in building the solution faster.



When you create a new module in the repository system, it creates a folder named dmc\_module to denotes the modules. The implementation of JAR files are then linked to the particular module in the module folder.

Life Cycles of Documentum

A lifecycle in Documentum states the different stages a document goes through during the ‘lifetime’ of that particular document. The lifecycle of Documentum gives an idea of how the changes are made in any document as it progresses through its lifetime. A lifecycle in Documentum also provides you with a notification indicating when the document gets approved by calling the custom codes and performing specific operations.

It can also automate the operations on a document for example if you want to reflect the approval\_date of the particular document, a lifecycle state extension can easily automate the operation.

Lifecycles are created and maintained for the usage in any document by the help of Documentum developer studio. Lifecycles are stored in dm\_policy as objects in the docbase. However, only one lifecycle can be attached to a particular document. The  r\_policy\_id attribute and the r\_current\_state attribute are maintained to store document’s lifecycle and its current state respectively.

What is meant by Web Services?

The DFC and BOF provide developers with the binding of custom made business logic’s that are referred to as the Web Services of Documentum. These custom logic are responsible and are used to call the functions on another document across the internet or in any organization i.e. intranet.

For example, SAP in any organization’s system can communicate with the help of Documentum web services. This functionality of a content management system provides you with a vast variety of knowledge in the form of data and information. This feature also provides you with a broader understanding of the organization such as the productivity or growth of any enterprise.

The main goal of web services is to provide a one-stop-shop for enterprises by combining a bunch of systems into one easy-to-use service and making the system mono-lingual i.e. one language SOAP (simple object access protocol). This way it allows us to understand multiple systems and their functionalities using only one interface.

In order to develop these web services over the Documentum API, developers have to work with their respective toolkits and applications. These web services are then implemented over the web servers providing their functionalities to desktop applications or any web applications.

Documentum Web-Based Applications

There are several Web-based applications that provide you with access to several different repositories. Some of these Web-Based applications are :

* Documentum WebTop
* Documentum Administrator (DA)
* Documentum xCP
* Documentum D2
* My Documentum

Documentum Microsoft Windows-Based Applications

The Documentum Microsoft Windows-based applications work only on the offline mode on the system. It also provides you the access for multiple repositories.

Some of the popular Documentum Microsoft Windows-based applications are mentioned below:

* **Documentum Desktop:** it also provides third-party integration to various Microsoft products  such as
  + windows explorer
  + Microsoft office
  + Microsoft outlook
* **Authoring Integration Services (AIS):** this is a Documentum system that works on both Windows and Macintosh systems.

Important Products of Documentum

Here are some of the important products of the Documentum system:

* Captiva IEC
* Captiva OEM
* xPression Enterprise Edition
* xPression Documentum
* Documentum Platform
* Documentum D2
* ApplicationXtender
* Documentum xCP
* Energy & Engineering Solutions

Documentum WebTop

EMC Documentum WebTop or web desktop is a Web-Based Documentum or client application. A WebTop consists of web services, web applications, client-server applications and many more. Web desktop is similar to the environment provided by the Mac OS,  Microsoft or Linux/UNIX based systems.

However, in Web Desktop the web applications, files, configurations, and the operations, all the functionalities are performed over the network. This provides you the access of the functionalities of Documentum remotely i.e. you can access, edit, delete or view the files anytime and from anywhere. WebTop provides an improved interface with more flexibility.

In any large or medium enterprise, the CMS systems have an excessive amount of workload such as publishing, reviewing or sharing a large amount of information. Web Desktops helps in eliminating this workload by providing numerous settings and specific roles to the group of users based on their work groups, memberships, and location within the enterprise. With the help of WebTop, you can exclude the unnecessary features and focus on more important and particular task resulting in increased productivity and faster processing speed.

It also provides you to customize and modify the system according to your requirements eliminating the dependencies on IT departments to perform manual coding changes. The WebTops give you access to numerous amount of widgets making navigation through the system much easier.

Documentum Web Publisher

Documentum Web Publisher is a web-based application that also provides the simplification of navigation through the system. Web publisher has the capability to automate the creation and publication of documents. It uses the Content Server of Documentum to store and process content into repositories and EMC’s Site Caching Services (SCS) of Documentum in order to publish the content to the web.

The Documentum Web Publisher provides you with the power of creating, designing and managing interactive portals and websites over the web. Any Non- Technical user can also avail the benefits of web publisher as it provides easy-to-use tools and widgets that help a user to create or edit the content as per their requirements.

It also helps you in understanding and accessing the websites which are containing multiple languages by providing advanced functionalities such as navigation, lifecycles, workflows and site structure.

Documentum Digital Asset Management (DAM)

EMC Documentum Digital Asset Manager (DAM) is a powerful software which provides the administration and intellectual abilities to a client for a content management system. This management system offers full leverage over clients digital assets by enabling the usage of complete sets of a content management system provided by the Documentum systems.

Digital Asset Management (DAM) also known as Media Assets Management (MAM), Rich Assets Management (RAM) System. DAMA system can be implemented via hosting software using product suits or can be installed via deployment.

**Some of the key features and benefits of DAM are mentioned below:**

* **File Transformations:** Transforms the digital media into multiple formats automatically making the content ready to use over any platform.
* **Thumbnail Display:** Provides the view of the files from any folder or search results in thumbnail and a magnified preview.
* **Active Preview:** It can read the document containing multi page with a web styled preview or the document preview.
* **Collections:** A collection of content can be created and managed by the help of this tool making the content ready to transform, or share or download in any specific formats.
* **Rendition/Relation Browsing:** It can also look and collect all type of versions of a document be it a low-resolution version file or web-ready version of the file.

Features of Documentum

Here are some of the key features of Documentum mentioned that makes this software different from other content management systems:

* **Regulatory compliance:**  Regulatory compliances are the guidelines that any organization have to follow. These rules are followed by the Documentum systems as it manages the content in compliance with great building practices.
* **Secure, centralized repository:** be it approval of content, reuse of content or reviewing, Documentum performs the operations by providing a secure and centralized repository.
* **Windows-based, web access:** Windows-based Documentum providing software to work over the documents in offline mode. Also providing the web-based system which provides content management over the web or internet.
* **Controlled life-cycle:**  Implement a lifecycle to every document in order to perform the operations on the content such as approval, reuse, and review.
* **Pure web-top extension:**  Provides advanced tools and user interface (UI) availing the features of extended search power and interactive widgets to navigate through the system.
* **Collaboration and retention support:**  It has provided collaborative content while also authoring the content along with retention control support.
* **Enhanced auditing:** Legal requirements and satisfactory system requirements are fulfilled to make sure the chain of command.
* **Compliance control:** impose a proper approval system with eSignatures to provide a compliance control system.

Benefits of Documentum

* Role-based security system
* Easy-to-use software with providing flexibility and faster navigation.
* Act as a platform and have an Ecosystem of complementary products like Imaging, Publishing and File Sync & share, etc.
* Reusability of the content
* Meet regulatory standards through features like Audit trail
* The organizations can save a lot of time, effort and cost.
* Highly configured search engine that enables the user to access the specific document from the system.
* Method to build/configure business rules around content like workflows and lifecycles.
* Provides both web-based as well as Windows-based system
* Companies can become more productive
* Provides better decision-making power by analyzing and producing high-quality content

Conclusion

Therefore, we have seen how Documentum can help in enhancing the growth of any enterprise. It ensures you with an effective content strategy that can match up with your document management requirements. It helps the organization in saving a lot of time, efforts and cost.

EMC’s Documentum system can be configured through multiple channels i.e. by email, printer, mobile, desktop and many more, making it remotely accessible. With Documentum, decision-making process in an enterprise can be more effective thus increasing the productivity of the company. Lastly, the Documentum system’s security protocols ensure that the documents are accessed and utilized by the right user while also following the regulatory compliances.