Windows Workflow Foundation

1. What is Windows Workflow Foundation (WF)?

Windows Workflow Foundation (WF) is a technology that was first introduced in .NET Framework 3.0. WF consists of a programming model, a workflow runtime engine, workflow designer, a rules engine, and tools to quickly build workflow-based applications on Windows. WF facilitates the separation between the business process code and the actual implementation code.

2. What are the components of WF 4.0?

WF consists of several components that work together to create desired workflow. The components of WF are given as follows:

- Workflows and activities
- Base activity library
- Custom activities
- Host process
- Activity data model!
- Runtime engine
- Runtime services

3. What is a workflow?

A workflow is a collection of actions (called activities) that presents the model of a process. A workflow provides a way to describe the order of the execution of a long running process and relationships between different activities. Multiple instances of a workflow may be active at any given moment in an application.

4. What are the different types of workflow in WF?

In WF 4.0, the following two types of workflows are used:

- **Flowchart workflows** - Helps you to create workflows using the common flowchart elements. In WF, the Flowchart activity is generally used to implement a non-sequential workflow, and occasionally it implements sequential workflows in case the FlowDecision nodes are not used. The Flowchart activity contains a collection of flow nodes, which inherit from the FlowNode class. The following types of nodes or elements can be a part of a flowchart:
  - **FlowStep** - Executes activities of a flowchart in a sequence.
  - **FlowDecision** - Shows the execution on the basis of a Boolean condition. It is similar to the If construct.
  - **FlowSwitch** - Shows the execution on the basis of an exclusive switch. It is similar to the Switch construct.

- **Procedural workflows** - Helps you to create workflows using basic and sequential execution standards. In WF, procedural workflows use flow control constructs, such as While, Switch, ForEach, and If, to execute activities. These flow control constructs are similar to those found in procedural languages. Procedural workflows can also contain other flow control activities, such as Flowchart and Sequence.

5. What are the four workflow principles?

According to Microsoft, there are four major principles that explain the behavior and working of workflows. Developers can use these principles while developing workflow-based applications. The four principles are as follows:
• Workflows help in coordinating the work performed by people and software.
• Workflows are long-running and stateful.
• Workflows are based on extensible models.
• Workflows remain transparent and dynamic throughout their lifecycle.

6. What is a base activity library?

The base activity library is a collection of activities used to create workflows.

7. What are XOML files?

WF provides developers a declarative way to create workflows by using extensible Application Markup Language (XAML). The files used to store such workflow markups are known as extensible Object Markup Language (XOML) files.

8. Can you integrate workflow applications with some other application, such as Windows Forms applications and Web applications?

Yes.

9. What is the difference between a system workflow and a human workflow?

A system workflow is a workflow that is developed to automate interactions among applications. Such workflow is usually static and predictable. On the other hand, a human workflow is a workflow that coordinates interactions of applications with people. As human workflows involve both software and people, they need to be more flexible than system workflows.

10. How can you implement a condition in a workflow?

You can implement a condition by using either of the following ways:

• By creating a rule condition - Specifies that you can implement conditions either directly in code or by using a tool, called the Rule Condition Editor. Rule conditions are stored in a separate Extensible Markup Language (XML) file. When a rule condition occurs in a workflow, the expression in a condition is evaluated and a Boolean value is returned.
• By creating a code condition - Refers to defining a condition directly in code. A code condition can be created by writing a method in the code. The method contains code for the condition and returns a Boolean value.

11. What is the function of the Rule Condition Editor dialog box in WF?

You can create and modify declarative rule conditions by using the Rule Condition Editor dialog box.

12. Explain the concept of Bookmarks in WF 4.0.

In WF 4.0, a bookmark is a mechanism that enables an activity to wait for an input without interrupting a workflow thread. When an activity signals that it is waiting for the input from a user, it can create a bookmark. A bookmark is created by using the BookmarkOptions class. This class provides the following bookmark types:

• None - Represents a bookmark that can be resumed exactly once. This is the default bookmark type.
• MultipleResume - Refers to a bookmark that you can resume multiple times.
• NonBlocking - Refers to a bookmark that does not block the functioning of the workflow.

13. What is a host process?
A host process is an executable program that hosts a workflow. It may be a Windows Forms application, a Web application, or a Web service application. You can use Web services in the host process or remoting to enable other applications to communicate with the workflow.

14. What are runtime services?

Runtime services consist of predefined and user-defined classes that are available to the workflow runtime engine during execution to customize the behavior of workflow runtime. Some of the runtime services available in WF 4.0 are as follows:

- **Scheduling services** - Enable creating and scheduling new workflow instances for execution.
- **Work batch services** - Enable behavior to maintain a stable and consistent execution environment.
- **Persistence services** - Enable you to save or restore the state of a running workflow for later use. You can restart the saved workflow anytime in future, even after weeks of inactivity.
- **Tracking services** - Enable you to monitor the state of the workflows. This is particularly useful when you have multiple workflows active at the same time (for example, in a shopping cart application).
- **Timer service** - Manages the timing required by the DelayActivity activity.
- **Transactions services** - Provide the transaction support needed for data integrity.
- **Data exchange services** - Manage custom communication services.
- **Threading services** - Administer physical threads used to execute workflow instances.

15. Which component of WF architecture is responsible to execute each workflow instance?

WF runtime engine is responsible to execute each workflow instance.

16. Which option do you need to select for the Condition property, if you want to create a code condition?

You can select the **Code Condition** option to create a code condition.

17. Explain Custom Activities.

In addition to the standard activities available within the base activity library, you can create new activities to meet specific business needs. Creating custom activities may be required to support a particular application that you want to integrate with WF. Custom activities are generally created through attributes and inheritance. You can create two types of custom activities, base and composite. You can create basic custom activity by inheriting the Activity class and custom composite activity by inheriting the compositeActivity class or a derived type.

18. What is a dynamic update?

Dynamic update is a powerful feature of WF that describes the ability of WF to modify the execution path of a running workflow. This feature is used in circumstances that call for extraneous behavior that was not modeled by the original workflow developer.

19. What is a runtime engine?

A runtime engine of WF provides the basic functionality to execute and manage the workflow lifetime. It runs within the host process and is responsible for executing each workflow instance. A host process can interact with multiple runtime engines at a time, where each engine executes multiple workflow instances. The host process interacts with runtime engine by using any of the following classes:

- **WorkflowInvoker** - Invokes a workflow as its method.
- **WorkflowApplication** - Controls the execution of a single workflow instance explicitly.
• **WorkflowServiceHost** - Hosts the workflows and allows sending and receiving messages among various instances of workflows.

20. What is an activity?

In Windows Workflow Foundation 4.0, an activity is the basic unit of composition and execution of a workflow. Each activity in a workflow consists of its own variables and arguments and is a subclass of the Activity class. These activities provide facilities for flow control, exception handling, data persistency, loading or unloading workflows, tracking, and transaction flow.

21. Explain why workflows are based on Extensible Models.

Workflows serve the purpose of automating business processes. Now, since each type of business has a wide range of problems; therefore, a workflow platform needs to be extensible. WF provides you with a set of base activities, such as IfElse, Code, and Delay, to build a workflow. You can extend these activities or build new activities to meet your requirements. Besides activities, you can also extend services, such as tracking, management, and persistence, provided by the runtime engine.

22. Write the steps that are involved in the sequential workflow, by default.

By default, a sequential workflow has only two steps:

- Start
- Finish