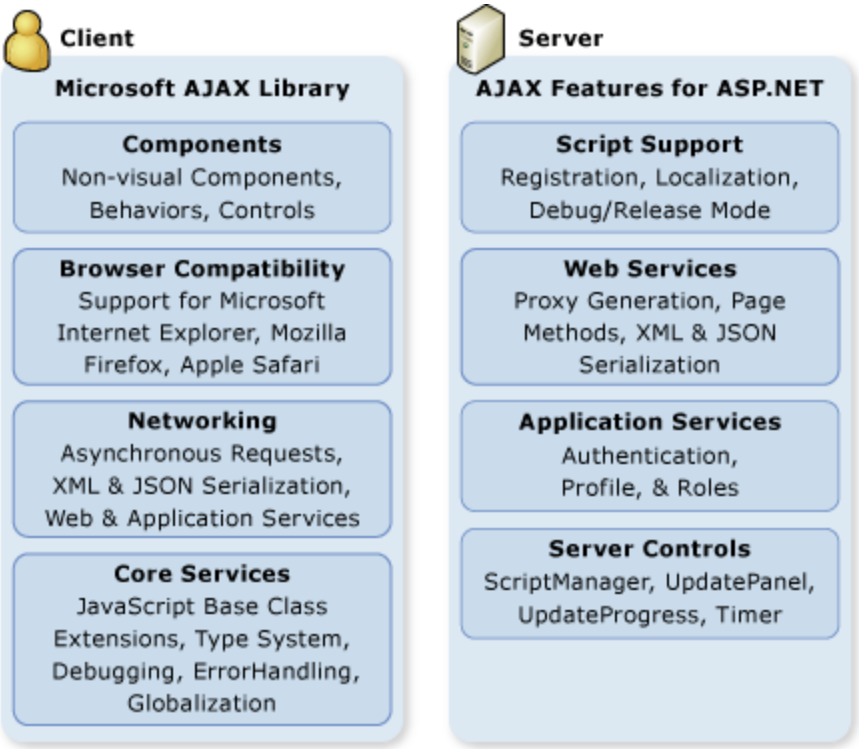
PART B

1.Explain about Ajax Client Server Architecture with Traditional Web Architecture.(N/D-16)(M/J-16)(N/D-2015)

A Microsoft Ajax Web application consists of either a client-only solution or a client and server solution. A client-only solution uses Microsoft Ajax Library but does not use any ASP.NET server controls. For instance, an HTML can include script elements that reference the Microsoft Ajax Library .js files. The Microsoft Ajax Library allows Ajax applications to perform all processing on the client. A client and server solution consists of using both the Microsoft Ajax Library and ASP.NET server controls.



server solution consists of using both the Microsoft Ajax Library and ASP.NET server controls.

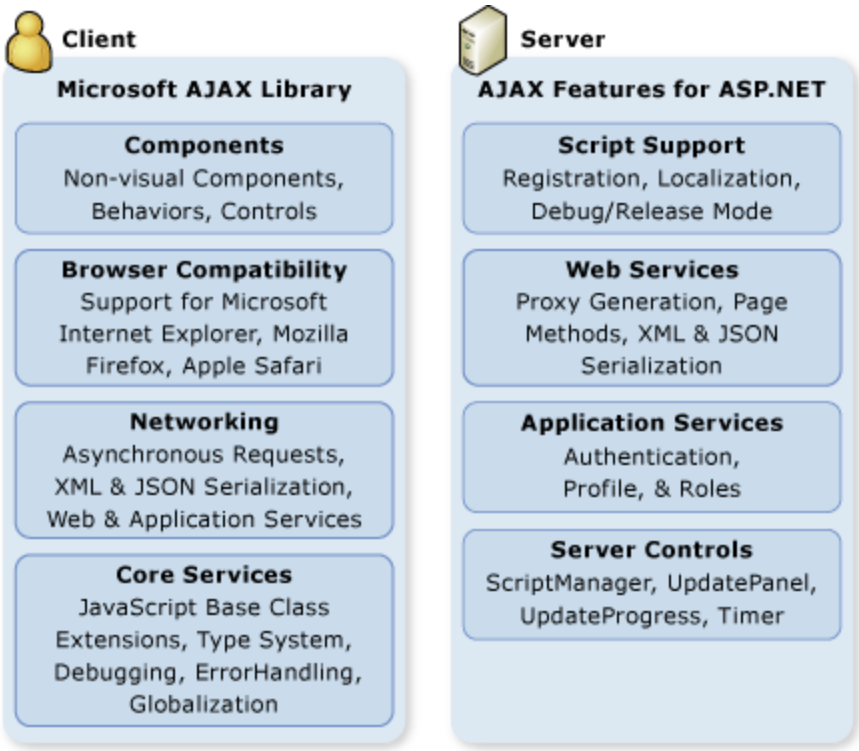


Figure: Microsoft Ajax client and server architecture

The illustration shows the functionality of the client-based Microsoft Ajax Library, which includes support for creating client components, browser compatibility, and networking and core services. The illustration also shows the functionality of server-based Microsoft Ajax features, which include script support, Web services, application services, and server controls. The following sections describe the illustration in more detail.

Microsoft Ajax Client Architecture

The client architecture includes libraries for component support, browser compatibility, networking, and core services.

Components

Client components enable rich behaviors in the browser without postbacks. Components fall into three categories:

- Components, which are non-visual objects that encapsulatecode.
- Behaviors, which extend the behavior of existing DOMelements.
- Controls, which represent a new DOM element that has custombehavior.

The type of component that you use depends on the type of client behavior you want. For example,a watermark for an existing text box can be created by using a behavior that is attached to the text box BrowserCompatibility

The browser compatibility layer provides Microsoft Ajax scripting compatibility for the most frequently used browsers (including Microsoft Internet Explorer, Mozilla Firefox, and Apple Safari). This enables you to write the same script regardless of which supported browser you

are targeting.

The networking layer handles communication between script in the browser and Web-based services and applications. It also manages asynchronous remote method calls. In many scenarios, such as partial-page updates that use the UpdatePanel control, the networking layer is used automatically and does not require that you write any code.

The networking layer also provides support for accessing server-based forms authentication, role information, and profile information in client script. This support is also available to Web applications that are not created by using ASP.NET, as long as the application has access to the MicrosoftAjax Library.

The networking layer handles communication between script in the browser and Web-based services and applications. It also manages asynchronous remote method calls. In many scenarios, such as partial-page updates that use the UpdatePanel control, the networking layer is used automatically and does not require that you write any code.

The networking layer also provides support for accessing server-based forms authentication, role information, and profile information in client script. This support is also available to Web applications that are not created by using ASP.NET, as long as the application has access to the Microsoft Ajax Library.

Core Services

The Ajax client-script libraries in ASP.NET consist of JavaScript (.js) files that provide features for object-oriented development. The object-oriented features included in the Microsoft Ajax client-script libraries enable a high level of consistency and modularity in client scripting. The following core services are part of the client architecture:

- Object-oriented extensions to JavaScript, such as classes, namespaces, eventhandling, inheritance, data types, and objectserialization.
- A base class library, which includes components such as string builders and extended error handling.
- Support for JavaScript libraries that are either embedded in an assembly or are provided as standalone JavaScript (.js) files. Embedding JavaScript libraries in an assembly can make iteasier to deploy applications and can help solve versioning issues. Debugging and ErrorHandling

The core services include the Sys.Debug class, which provides methods for displaying objects in readable form at the end of a Web page. The class also shows trace messages, enables you to use assertions, and lets you break into the debugger. An extended Error object API provides helpful exception details with support for release and debug modes.

Globalization

The Ajax server and client architecture in ASP.NET provides a model for localizing and globalizing client script. This enables you to design applications that use a single code base to provide UI for many locales (languages and cultures). For example, the Ajax architecture enables JavaScript code to format Date or Number objects automatically according to culture settings of the user's browser, without requiring a postback to the server.

Ajax Server Architecture

The server pieces that support Ajax development consist of ASP.NET Web server controls and components that manage the UI and flow of an application. The server pieces also manage serialization, validation, and control extensibility. There are also ASP.NET Web services that enable you to access ASP.NET application services for forms authentication, roles, and user profiles.

Script Support

Ajax features in ASP.NET are commonly implemented by using client script libraries that perform processing strictly on the client. You can also implement Ajax features by using server controls that support scripts sent from the server to the client.

You can also create custom client script for your ASP.NET applications. In that case, you can also use Ajax features to manage your custom script as static .js files (on disk) or as .js files embedded as resources in an assembly.

Ajax features include a model for release and debug modes. Release mode provides error checking and exception handling that is optimized for performance, with minimized script size. Debug mode provides more robust debugging features, such as type and argument checking. ASP.NET runs the debug versions when the application is in debug mode. This enables you to throw exceptions in debug scripts while minimizing the size of release code.

Script support for Ajax in ASP.NET is used to provide two important features:

- The Microsoft Ajax Library, which is a type system and a set of JavaScript extensions that provide namespaces, inheritance, interfaces, enumerations, reflection, and additionalfeatures.
- Partial-page rendering, which updates regions of the page by using an asynchronous postback. Localization

The Microsoft Ajax architecture builds on the foundation of the ASP.NET 2.0 localization model. It provides additional support for localized .js files that are embedded in an assembly or that are provided on disk. ASP.NET can serve localized client scripts and resources automatically for specific languages and regions.

Web Services

With Ajax functionality in an ASP.NET Web page, you can use client script to call both ASP.NET Web services (.asmx) and Windows Communication Foundation (WCF) services (.svc). The required script references are automatically added to the page, and they in turn automatically generate the Web service proxy classes that you use from client script to call the Web service.

You can also access ASP.NET Web services without using Microsoft Ajax server controls (for example, if you are using a different Web development environment). To do so, in the page, you can manually include references to the Microsoft Ajax Library, to script files, and to the Web service itself. At run time, ASP.NET generates the proxy classes that you can use to call the services.

Application Services

Application services in ASP.NET are built-in Web services that are based on ASP.NET forms authentication, roles, and user profiles. These services can be called by client script in an Ajax-enabled Web page, by a Windows client application, or by a WCF-compatible client.

Server Controls

Ajax server controls consist of server and client code that integrate to produce rich client behavior. When you add an Ajax-enabled control to an ASP.NET Web page, the page automatically sends supporting client script to the browser for Ajax functionality. You can provide additional client code to customize the functionality of a control, but this is not required.

2. Show the relationship between SOAP, UDDI, WSIL and WSDL

The following standards play key roles in Web services:

•Universal Description, Discovery and Integration (UDDI),

•Web Services Description Language (WSDL),

•Web Services Inspection Language (WSIL), SOAP, and

•Web Services Interoperability (WS-I).