**Advanced Web Programming**

1. **Describe the anatomy of XSL-FO namespace.**

Ans. As specified earlier namespaces identify the particular markup vocabulary from which element-type names and attribute names are derived. They are important because, for example, when using links, the idea of embedding linked content from other documents raises the problem of possible duplicate element-type names and attribute names with different meanings. For instance, if the element-type name “body” was used in two different ways, such as by an auto parts manual and by a physician's desktop reference. A namespace is, well, a space for a name. Element-type names or an attribute names are qualified with a prefix, a sort of short name or alias for the namespace. Each namespace is declared prior to using its prefix, either in the same element in which it is used or in an ancestor element. The declaration of the namespace contains a URI that (hopefully) points to something that regulates that namespace. A namespace declaration is similar to a DOCTYPE declaration, only it is found in a different part of the XML document instance. It is not a child of the root; it is an attribute on an element.

**Anatomy of a namespace**: A namespace must be declared before it can be used. Declaring a namespace such as xmlns:xsl="http://www.w3.org/1999/XSL/Transform [http://www.w3.org/1999/xsl/transform]" defines a prefix and the value of the namespace name, which is usually a URI pointing to the namespace owner. The prefix (xsl, in this example) follows the XML namespace declaration, xmlns. The XML namespace declaration is the reserved attribute name, followed by the: separator. Immediately following the separator is the prefix being defined, followed by an equal sign, which separates it from the address or URI of the namespace. The URI address of the namespace is only used to identify the namespace; whether it resolves to a location or a schema is irrelevant.

1. **What are the advantages of XML?**

Ans. There are many advantages of XML. To exchange information use of XML offers many benefits.

* XML can store and organize just about any kind of information in a form that is tailored to your needs.
* As an open standard, XML is not tied to the fortunes of any single company, nor married to any particular software.
* With Unicode as its standard character set, XML supports a staggering number of writing systems (scripts) and symbols, from Scandinavian runic characters to Chinese Hanideographs.
* XML offers many ways to check the quality of a document, with rules for syntax, internal link checking, comparison to document models, and data typing.
* With its clear, simple syntax and unambiguous structure, XML is easy to read and parse by humans and programs alike.
* XML is easily combined with style sheets to create formatted documents in any style you want. The purity of the information structure does not get in the way of format conversions.

All of these advantages come at a time when the world is ready to move to a new level

of connectedness. The volume of information within our reach is staggering, but the limitations of existing technology can make it difficult to access. Businesses are scrambling to make a presence on the Web and open the pipes of data exchange, but are hampered by incompatibilities with their legacy data systems. The open source movement has led to an explosion of software development, and a consistent communications interface has become a necessity. XML was designed to handle all these things, and is destined to be the grease on the wheels of the information infrastructure.

1. **What is Web Services Description Language (WSDL)? Explain.**

Ans. Web Service Description Language (WSDL) is a structured way to describe all web service related issues. When a WSDL file is complete, it contains enough information that an application such as an integrated development environment (IDE) can automatically generate classes and applications that access the service. It is considered as another advantage of web services where they can be described formally in a document which is known as WSDL.

Like SOAP it is also an XML document. In addition, it is authored with a specific XML grammar that was devised to communicate metadata about a Web service in a uniform manner to all potential clients. In other words, a WSDL document is an XML document that conforms to a specification. All the metadata about the Web service is contained somewhere in this file. Let’s look into the structure of a WSDL document.

A WSDL document consists of a hierarchy of descriptions, each more granular than the one before. These decryptions define three fundamental properties of a Web service:

**What a service does:** The operations (methods) the service provides, and the data (arguments and returns) needed to invoke them.

**How a service is accessed:** Details of the data formats and protocols necessary to access the service’s operations.

**Where a service is located:** Details of the protocol-specific network address, such as a URL.

1. **Discuss following XML publishing Methods**
2. **Use SQL/XML:** SQL/XML is an ANSI and ISO standard that provides support for using XML in the context of a relational database management system. Because SQL is the standard language for accessing and managing data stored in relational databases, it is only natural that enterprises and users want the ability to (access and) generate XML data from their relational data through the use of the SQL interface. SQL/XML defines a standard that makes it possible to store your XML documents in your SQL database, to query those documents using XPath and XQuery, and to "publish" your existing SQL data in the form of XML documents.

Currently the following SQL/XML publishing functions are available in DB2 for z/OS Version 8:

* XMLELEMENT and XMLATTRIBUTES to construct XML elements with attributes.
* XMLFOREST to construct a sequence of XML elements.
* XMLCONCAT to concatenate XML elements.
* XMLAGG to aggregate XML elements.
* XMLNAMESPACES to declare one or more XML namespaces.
* XML2CLOB is not part of the SQL/XML standard. This function is

Introduced by DB2, because currently (Version 8), DB2 does not support a native XML external data type. Therefore, the internal XML data type that is used inside the DB2 engine needs to be transformed to a data type (CLOB) that DB2 understands and is able to return to the application. Once the native XML data type becomes available, there will be no need for this function anymore.

1. **Use DB2 XML Extender:** IBM DB2 is a family of database server products developed by IBM. These products all support the relational model, but in recent years some products have been extended to support object-relational features and non-relational structures, in particular XML. You can also use DB2 XML Extender to generate XML documents from data in relational tables, either in an XML Extender XML Collection, or just in a set of regular DB2 tables. During publishing, XML Extender also uses a DB2 to map the relational tables to the XML structure.
2. **Explain any five new features of HTML5 that are not in HTML.**

Ans. There are new features of HTML5. Let’s learn the use of new elements in HTML5 to design web pages. HTML5 provides more 20 new elements to enhance our web pages adding semantics to deliver more accessible, reusable content. New HTML5 elements can be categorized as:

|  |  |
| --- | --- |
| **Category** | **Elements** |
| Semantic element for  structuring a page | <article>, <aside>, <figcaption>, <figure>, <footer>,  <header>, <hgroup>, <nav>, <section>  <details>, <summary> |
| Audio, video, and  plug-ins | <audio>, <video>, <source>  <embed> (previously supported, but now an official  part of the language) |
| Canvas | <canvas> |
| Semantic elements  for text | <mark>, <time>  <wbr> (previously supported, but now an official  part of the language) |
| Web forms and  interactivity | <input> (not new, but has many new subtypes)  <datalist>, <keygen>, <meter>, <progress>,  <command>, <menu>, <output> |

1. **Define MIDlet and describe MIDlet user interfaces.**

Ans. **MIDlet suites:** We know MIDlet is simple java program intended for mobile device,similarly MIDlet suites is collection of one or more MIDlets packed togetherusing a Java Archive (JAR) file. Actually the MIDlet is subject to some rulesregarding its run-time environment and packaging. In terms of run-time environment the application manager starts the MIDlet. By starting MIDletthe application manager accesses following resources:

* All files contained within the application’s JAR file, which contains all the classes required to run the application, along with all the resources, such as image files and level data.
* The contents of the MIDlet descriptor file. Application execution options can be set as properties with it.
* Classes made available as part of the CLDC and MIDP libraries. Multiple MIDlet applications can be bundled together within one JAR file.

Now if we are developing a more sophisticated MIDlet that contains many classes, images, application parameters, and soon, you need to package your MIDlet into a MIDlet suite. Packaging creates a JAR file containing all your class and resource files (such as images and sounds) and the application descriptor (JAD) file.

A JAD file includes MIDlet-name, MIDletversion, MIDlet-Vendor (MIDlet suite provider, organization), MIDlet-Jar-URL, MIDlet-Jar-size.

**MIDlet User Interfaces:** We must have understandthat MIDlets are intended for a huge range of devices ranges from small twocolor devices to multicolour display with touch screen. Creating userinterfaces for such a wide range of devices is a very critical job. One optionavailable to the MIDP designers was to use a subset of the AbstractWindows Toolkit (AWT) or Swing components from J2SE. Designing a userinterface for Java ME applications on mobile devices is quite a challengingtask, because MIDP’s flexibility allows it to be used in hundreds of differentdevice models, with different form factors, hardware, screen sizes and inputmethods. MIDP includes two types of API: “High level” API and “Low level”API. The term high-level refers to the API's high level of granularity that isprovided to the programmer in two areas:

* ability to manipulate the look and feel of UI widgets
* granularity of information about events and event handling

The low level API gives the developer exactly the opposite of the high level API. It gives following access:

* Complete control over screen, means writing code to draw everything that appears on the user's screen.
* Access to the key pad which means interpreting every input keystroke and pointer movement to decipher what the user wants to do.

The MIDlet user interface library, which is implemented in the javax.microedition.lcdui package, includes several classes that represent the device's screen and provide the basic top-level windows. Developers can use these as the basis for building form-based MIDlets or more graphically sophisticated MIDlets, such as games. The LCDUI toolkit is a set of features for the implementation of user interfaces especially for MIDPbased devices.