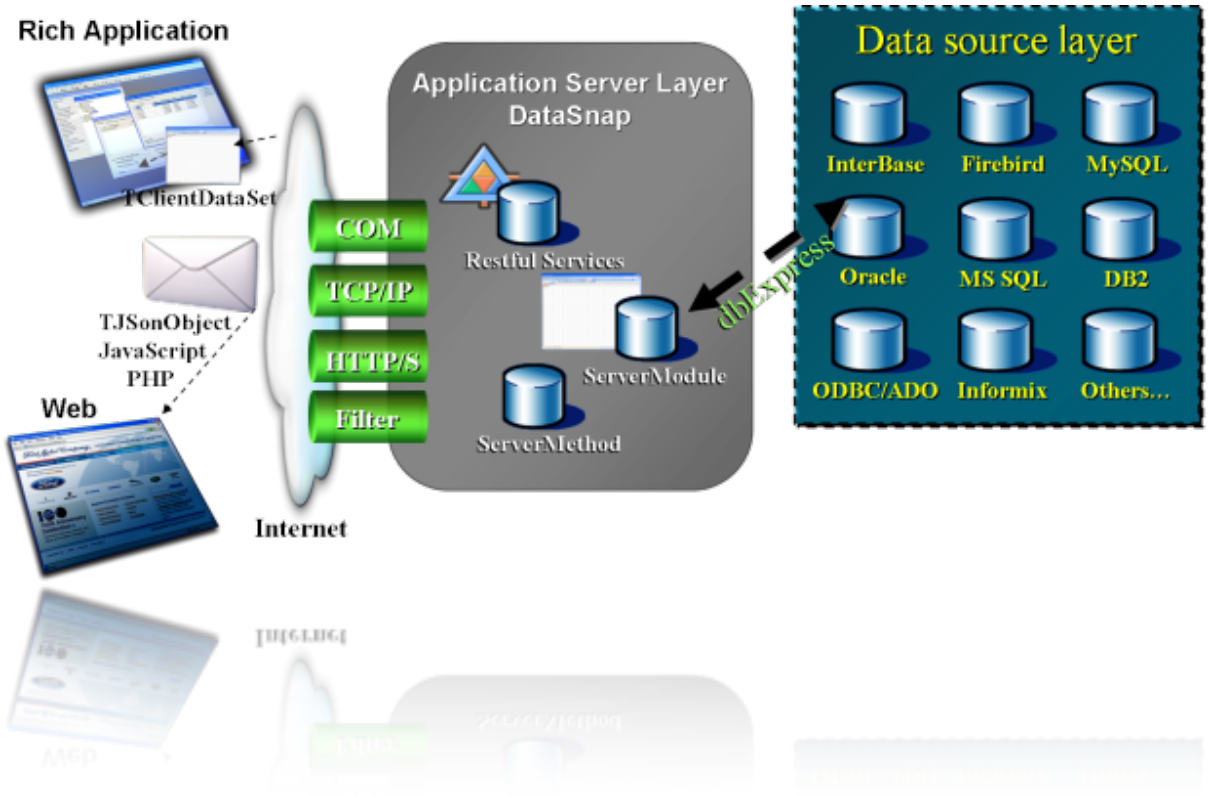
Contenido del Material

Introducción

La tecnología DataSnap sigue evolucionando, por el aumento en la demanda de aplicación distribuidas. La tecnología detrás de DataSnap se ha movido del enfoque remoto soportado por la plataforma Microsoft COM/DCOM a un enfoque de comunicación más abierta basada en TCP/IP. Esta evolución ha permitido que la tecnología DataSnap pudiese ampliar sus capacidades con la finalidad de soportar una capa intermedia completa. Una de las principales características de la tecnología es que es mucho más rápido: rápido para desarrollar, rápido de implementar y rápido y fácil de ejecutar o colocar en la producción.

La ampliación de las capacidades de DataSnap, nos permite desarrollar aplicaciones basadas en estándares. DataSnap por compatibilidad mantiene el soporte a COM/DCOM, además ahora tiene la capacidad de comunicarse de forma nativa a través de TCP/IP, alternativamente, a través de HTTP o HTTPS. Al mismo tiempo, la lógica del negocio que se encuentra en los servidores DataSnap, pueden ser despachados a los clientes como servicios RESTful.



Descripción y Arquitectura de DataSnap

Descripción:

Anteriormente conocido como MIDAS, DataSnap Delphi es una tecnología que permite el desarrollo de aplicaciones multi-capas, especialmente aplicaciones multi-capas de base de datos. También hay soporte completo para C++ Builder. DataSnap es independiente de MIDAS. DataSnap ofrece la posibilidad de crear aplicaciones cliente-servidor que se comunican a través de Internet, redes locales y servidores dedicados.

La característica principal de DataSnap es la capacidad que tiene la aplicación cliente para invocar métodos que se implementan en un servidor. DataSnap genera automáticamente las interfaces necesarias, para que el cliente pueda comunicarse con el servidor que contiene los prototipos de los métodos remotos.

DataSnap proporciona una forma para que el Cliente pueda comunicarse de forma segura con el servidor, mediante una transferencia de datos de manera segura en formato JSON (JavaScript Object Notation) a través de TCP/IP o HTTP(S). Permite definir filtros en ambos extremos del canal de comunicación, entre ellos: cencriptación y

compresión de la data transferencia, con el propósito de mejora la seguridad y el rendimiento en la transferencia, además permite crear nuestros propios filtros.

Otro de los beneficios de la tecnología DataSnap es que ofrece la posibilidad de notificar de forma asíncrona a todas las aplicaciones clientes conectadas de los cambios realizados en el servidor, de modo que los clientes puedan tomar las acciones apropiadas, ésto se logra a través de la implementación de "callbacks".

Arquitectura:

La tecnología DataSnap permite el desarrollo de aplicaciones cliente-servidor, donde el cliente o el servidor pueden ser desarrollados en Delphi o C++Builder. La comunicación entre el cliente y el servidor se pueden hacer en el mismo equipo, en una red local, o a través de la Internet. Usted puede elegir el tipo de conexión estableciendo la propiedad `CommunicationProtocol` de un componente `TSQLConnection` a TCP/IP o HTTP(S), en tiempo de diseño.

La comunicación entre cliente y servidor se realiza a través del intercambio de contenidos de datos JSON (JavaScript Object Notation). La respuesta del servidor también se pueden enviar en formato HTML, utilizando la tecnología WebBroker.

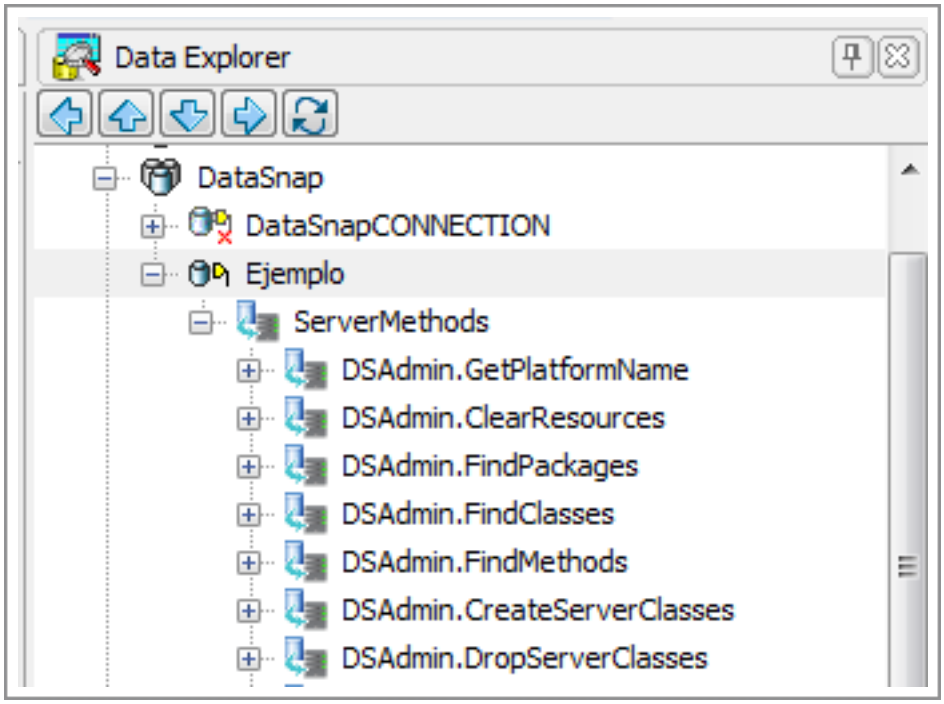
Si un cortafuegos está en ejecución en el cliente o en el servidor, DataSnap ofrece la opción de enrutar la transferencia de datos a través de un túnel, anulando el cortafuegos, permitiendo de esta manera la comunicación entre el cliente y el servidor.

DataSnap permite la implementación del cliente y del servidor en la misma aplicación, también conocidad como conectividad "in-process".

Visualizando de Métodos Remotos usando "DataExplorer":

Para utilizar DataExplorer con DataSnap, primero es necesario tener un servidor de aplicaciones DataSnap ejecutándose. Una vez que esta aplicación de servidor está en funcionamiento, vaya a la pestaña DataExplorer en la interfaz principal de Delphi.

En la ventana DataExplorer, en la sección DataSnap, nos damos cuenta de que podemos visualizar la categoría: "ServerMethods", tal como se muestra en la imagen siguiente:



Ejemplo de DataSnap Simple (Como en Primero)

Principales Componentes del Servidor DataSnap:

- TDSServer: El componente TDSServer es el corazón lógico del servidor DataSnap. Contiene métodos Start y Stop para iniciar y detener el servidor. También contiene la propiedad AutoStart. Por defecto el valor de AutoStart es True, por lo que el servidor se inicia automáticamente cuando la aplicación se ejecuta. Solo hay un componente TDSServer por aplicación de servidor.

- TDSServerClass: representa una clase de servidor. Al igual que TDSServer es necesario un componente TDSServerClass para proveer la comunicación entre el servidor y el cliente. El servidor DataSnap automáticamente crea y destruye instancias de clases del servidor. La creación de instancias de una clase de servidor se controla mediante la propiedad LifeCycle del componente TDSServerClass. La propiedad LifeCycle tiene tres valores posibles: Server, Session, e Invocación.
  - Server significa que el Servidor DataSnap crea una sola instancia de la clase del servidor, para ser usada por todos los clientes conectados, esto representa un

patrón de diseño "Singleton". Tenga especial cuidado cuando use la propiedad LifeCycle como Server, porque la implementación de la clase del lado del servidor necesita ser "thread-safe", usted debería diseñar esta clase para que pueda ser accedida por múltiples "threads".
  - Session es el valor por defecto de la propiedad LifeCycle, significa que el Servidor DataSnap crea una clase de servidor por cada cliente conectado.
  - Invocation en este caso una clase de servidor es creada y destruida cada vez que el cliente invoca un método, por tal motivo el estado de la clase de servidor no se mantiene entre la llamada de los métodos.

- TDSTCPServerTransport: este componente implementar el servidor socket "multithread" esperando por las conexiones de los clientes. Este componente no tiene eventos. La propiedad Port indica el puerto TCP en el cual él esta escuchando, por defecto tiene el valor 211. Es posible usar HTTP(S) para comunicarse entre los sevidores y los clientes DataSnap

Principal Componente del lado del Cliente:

- TSQLConnection: encapsula una conexión dbExpress a un servidor de Base de Datos. Para el caso de hacer una conexión a un Servidor DataSnap, sólo debemos configurar las siguientes propiedades:
  - Driver como DataSnap (Desde el punto de vista del cliente es una conexión a una Base de Datos, pero en realidad es una conexión a un servidor DataSnap).
  - LoginPrompt como False para evitar que aparezca el díalogo de usuario/ contraseña cada vez que el cliente se conecte.

Ver Video en el siguiente link: http://www.youtube.com/watch?v=JFONiqdj0t4