

Informe

Opus™ WebService

Personalización y presentación en tiempo real de documentos Adobe®
PDF

*Elixir crea
soluciones inte-
grales
para administrar
los
documentos que
reflejan su nego-
cio.*



Elixir Technologies Corporation
721 East Main Street
Ventura, California 93001 USA
+1 805 641 5900 voice
www.elixir.com

Copyright © 2006 Elixir Technologies Corporation. All rights reserved.

Contenido

Introducción	3
Industrias que usan Opus WebService	5
El panorama técnico	6
Ventajas técnicas	7
Ejemplos de uso	11

Los nombres de productos usados son marcas registradas de sus respectivos dueños.

Introducción

Servicios Web son una manera estandarizada de integrar aplicaciones basadas en la Web usando lenguajes como XML, SOAP, WSDL y UDDI en Internet. Permite que diversas aplicaciones se comuniquen entre ellas sin tener que realizar codificación innecesaria, por medio de comunicación XML; tampoco deben trabajar con algún lenguaje de programación o sistema operativo en particular.

Un Servicio Web crea una aplicación accesible por medio de protocolos estándares de Internet Como componentes, su funcionalidad puede ser reutilizada fácilmente. A diferencia de muchas tecnologías que son accedidas por medio de protocolos propietarios,

este es do vía los de como usando tos de univer- como

accesa-
protoco-
Internet
HTTP
formatos
sales
XML.



En términos prácticos los Servicios Web han emergido como mecanismo de integración de diferentes sistemas informáticos. Trabajan usando tecnologías ampliamente aceptadas, pueden ser adoptados parcialmente con poco riesgo y a bajo costo. Hoy las empresas utilizan los Servicios Web para la integración de aplicaciones punto a punto, para reutilizar elementos informáticos existentes, y conectarse con seguridad con sus socios comerciales o clientes. Los proveedores de software independientes incluyen la funcionalidad de Servicios Web en sus productos haciéndolos más fáciles de integrar.

Opus WebService es un ejemplo de software usado en el ambiente de Servicios Web.

Opus WebService proporciona una poderosa solución para la presentación en línea de documentos PDF y de impresión On Demand. Los clientes tienen acceso al sistema mediante un explorador Web o una aplicación, de donde pueden pedir un documento. El PDF se crea en tiempo real y se presenta en línea para su visualización o impresión.

Industrias usando Opus WebService

La presencia Web de productos permite a las organizaciones ampliar sus aplicaciones al Internet así como a Intranets o Extranets. Con sus aplicaciones en la Web, las organizaciones permiten su uso a sus clientes, proveedores, socios y/o empleados.

A continuación se presentan algunos escenarios para Opus WebService:

- Proveedores de servicios de impresión

Opus WebService puede ser usado para crear un flujo de trabajo de impresión en la Web. Clientes pueden cargar los diseños de sus documentos, gráficos y datos vía Opus WebService y crear una vista previa en tiempo real en la Web. Este acercamiento interactivo para realizar pruebas permite a los clientes experimentar con varios diseños de documentos y datos variables sin la ayuda del proveedor de servicios de impresión. Una vez que la prueba es aceptada el proveedor de servicios de impresión produce el documento

- Aplicaciones para Call center/help desk

Los centros de la ayuda que asisten a clientes con preguntas sobre documentos impresos pueden utilizar Opus WebService como método para visualizar el documento en pantalla exactamente como se vería el documento impreso. De este modo el cliente puede mirar su estado de cuenta y la persona de soporte puede ver el mismo documento en su pantalla en formato PDF.

El panorama técnico

Los diseños del Opus se pueden utilizar alternativamente para impresión o presencia Web.

Todos los diseños se pueden almacenar en un servidor y el servidor de Opus WebService puede cargarlos por requerimiento para la composición de un documento en particular. Las aplicaciones y recursos se pueden compartir con la instalación de Opus sin empobrecer el rendimiento. Sin embargo en una configuración más grande, debemos separar nuestros procesos batch y nuestro servidor de WebService, esto es necesario para tener una disponibilidad óptima del servidor Opus WebService.

El entorno de desarrollo para la impresión y la presencia Web puede ser idéntico. Un usuario típico desarrollaría sus diseños de Opus y los probaría en un servidor de la prueba. Después de corroborar que la prueba es exitosa, los diseños serían movidos al servidor de la producción (para su impresión) y al servidor de Opus WebService (para la presentación Web).

Los recursos tales como fuentes, formas e imágenes se pueden almacenar por separado para cada ambiente o pueden residir en un servidor común. Se recomienda que los recursos sean guardados separadamente si se utilizan imágenes de alta y baja resolución para la impresión y la presentación Web respectivamente. Las imágenes de baja resolución se recomiendan para Opus WebService ya que no aumentan el tamaño del archivo PDF y mantienen un rendimiento óptimo.

System Requirements

Los requisitos mínimos del hardware/software para Opus WebService son:

Pentium IV 2+ MHz

512 MB RAM

30 GB HDD

Windows 2003 Server con el último service pack

La configuración recomendada es:

Dual Pentium IV 3+ GHz Hyper Threaded processors

1 GB RAM por CPU

60 GB HDD

Fuerzas técnicas

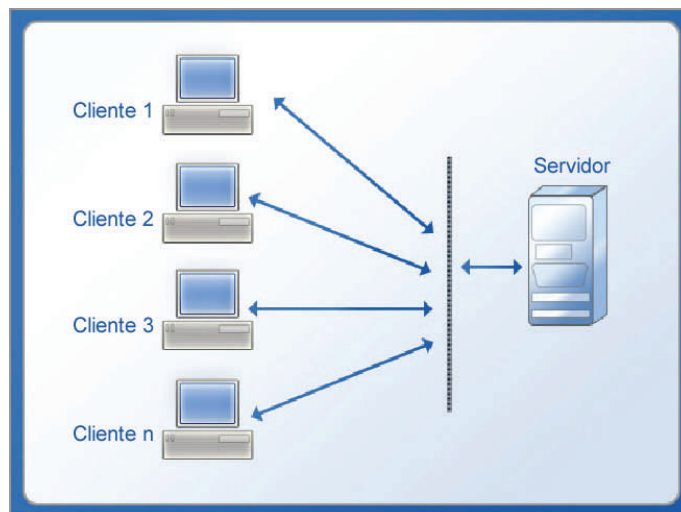
Opus WebService proporciona las cualidades técnicas necesarias para el correcto funcionamiento de sistema.

- Gran disponibilidad
- Rendimiento y escalabilidad
- Tolerancia por falla
- Capacidad de integración

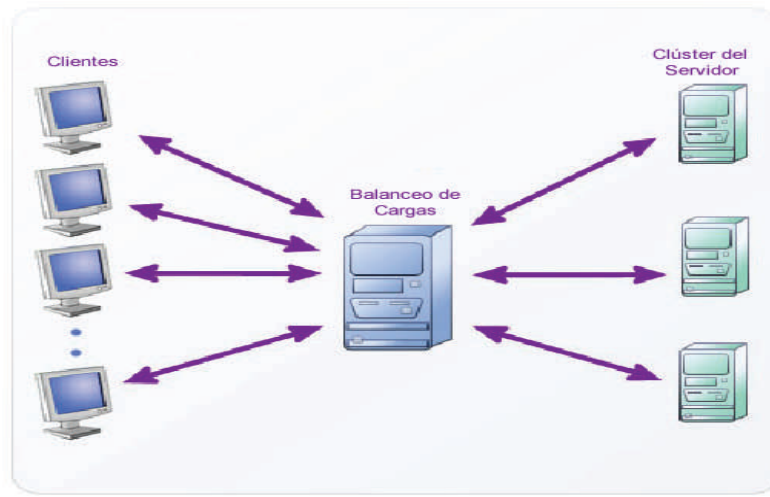
La gran disponibilidad de Opus WebService se basa en su tolerancia y escalabilidad. Esto permite que trabaje las 24 horas del día. El alto rendimiento se asegura con la integración del servidor en un ambiente de balanceo de cargas en conjunto con características escalabilidad.

La carga esperada en el sistema aunada al flujo de trabajo en línea depende del tamaño de la compañía. La carga máxima puede alcanzar un número muy grande de documentos procesados en un breve periodo de tiempo. La configuración de Opus WebService debe permitir al usuario final obtener una respuesta en menos de 5 segundos. Puesto que el funcionamiento de un solo *formatter* de Opus es limitado, esta solución permite el uso de varios *formatters*. Estos pueden funcionar en el servidor.

Configuración básica de Opus WebService



Configuración de Opus WebService, servidores múltiples



En el caso de que los servidores estén sobrecargados por exceso de peticiones sin un *formatter* Web disponible, la petición se almacenará y liberará según su orden.

La tolerancia por falla se creó para permitir que Opus WebService funcione en computadoras múltiples simultáneamente. Si una computadora falla, se utiliza automáticamente a un servidor disponible.

La interface de los Servicios Web permite que Opus WebService sea integrado con otras aplicaciones que funcionen en prácticamente cualquier plataforma (Windows, UNIX y Linux), enriqueciendo a Opus WebService con una amplia gama de capacidades de integración. El cliente puede llamar a Opus como un RPC (Remote Procedure Call) aunque Opus funcione en un servidor basado en Windows.

- Opus WebService se ser invocado sincrónicamente o asincrónicamente por clientes que usan lenguajes de programación tales como Java, C++, Visual Basic y C #.
- Opus WebService se puede invocar también usando códigos para Internet tales como ASP, ASP.NET, JSP y Java.

La licencia del paquete de producción Opus WebService está basada en el número de procesadores físicos en un servidor. El paquete de producción Opus WebService consiste en un diseñador de documentos y múltiples Web Formatters en razón del número de procesadores físicos y del PDF utilizado para presencia Web.

Tecnología base

La piedra angular de los Servicios Web es la adopción de XML porque su alto grado de uso en este tipo de sistemas. El valor del XML tomó forma cuando fue utilizado para conectar dos computadoras, dar formato a mensajes en XML y enviar estos mensajes usando protocolo HTTP. Así, el XML RPC (Remote Procedure Call) fue concebido. Además, se han definido muchos otros estándares y la tecnología actual de Servicios Web se basa en tres pilares principales:

- SOAP
- WSDL
- UDDI

SOAP (Simple Object Access Protocol) es un protocolo ligero para el intercambio de información en un ambiente descentralizado. Puede ser utilizado con otros protocolos como HTTP y FTP. La solución más adoptada es HTTP debido al hecho de que SOAP amplía el protocolo original del XML RPC. SOAP se utiliza actualmente en la mayoría de los productos de comunicaciones para Servicios Web que se pueden llamar de varios lenguajes computacionales en diferentes plataformas. Lo que es importante sobre SOAP es el hecho de que está orientado al objeto, es decir se utiliza para enviar mensajes a los objetos, que tienen una interfaz. SOAP puede ser usado para hacer RPC, que es un método básico de invocación usada en la Red, pero tiene características más avanzadas que permiten implementar diseños para el envío de mensajes unidireccionales o intercambio de mensajes.

La escritura de una aplicación que pueda invocar Servicios Web requiere información detallada sobre de la interfaz de dicho Servicio Web, métodos que soporta, parámetros y valores de retorno. WSDL (Web Services Description Language) facilita este requisito ya que es un lenguaje basado en XML con etiquetas y estructura predefinidas. WSDL describe un Servicio Web por medio de su interfaz, sus métodos y parámetros.

WSDL simplemente describe interfaces del objeto. No contiene ninguna información sobre la semántica; es decir la manera en que los métodos deben ser utilizados o cuáles los parámetros correctos, cuál es el propósito del objeto o información alguna de donde encontrarlo.

UDDI (Universal Description, Discovery, and Integration) es otro servicio basado en XML. UDDI funge como repositorio universal que almacena la información sobre Servicios Web. Contiene información elemental sobre el proveedor de Servicios de Web, interfaces y ubicación de la información. Por lo tanto, Servicios Web pueden ser utilizados para encontrar otros Servicios Web en la red. UDDI también soporta algunas descripciones básicas que son usadas para la especificación de la semántica del objeto. Para poder publicar Servicios Web a un mayor número de gente, repositorios públicos de UDDI han sido creados por compañías como Microsoft, IBM, NTT, que dan la oportunidad a proveedores de publicar la información sobre Servicios Web (véase www.w3c.org).

Ejemplos de aplicaciones

Industria: Educación

Aplicación: Migración a boletas distribuidas en la Web

En nuestra investigación encontramos que muchas organizaciones educativas tienen un proceso similar para distribuir sus boletas. Este ejemplo se basa en una compañía particular de nuestra investigación.

Proceso típico en la industria de hoy

El departamento de educación realiza exámenes y pasa los resultados a una compañía que proporciona boletas a los estudiantes. Esta compañía escanea los exámenes y almacena los resultados en una base de datos de Oracle. La compañía cuenta con personal para atención al cliente que procesa los pedidos de boletas.

Cuando una petición se recibe se envía al departamento de informática donde se ejecuta un código en COBOL para combinar los datos del estudiante con las formas pre-impresas de las boletas. El informe después se imprime y se envía al estudiante. Este proceso tarda típicamente 3 a 5 días antes de que el estudiante consiga los resultados pedidos.

Mejora del proceso

Moviendo el proceso a la Web el proceso se puede reducir a menos de un minuto.

Los estudiantes se registrarían en un sitio Web y pedirían sus resultados. El estudiante realiza su petición y en segundos ve un PDF con lo que pidió. El estudiante puede salvar el archivo o imprimirlo en una impresora local.

El nuevo proceso nos da acceso a las boletas inmediatamente, elimina la necesidad de los programadores de COBOL y reduce la carga de trabajo al departamento de atención al cliente.

Beneficios

- *Los informes se pueden generar sin la programación en COBOL*
- *Los estudiantes pueden ver sus resultados inmediatamente*
- *Los costos de envío se redujeron significativamente*

Ejemplos de aplicaciones

Industria: Proveedores de servicios de impresión

Aplicación: Cambio del proceso de prueba a la Web

Hacer outsourcing de la mayoría o la totalidad de la impresión de una compañía puede ser beneficioso considerando el alto costo de comprar y de mantener las impresoras de alto volumen.

Elixir cuenta varios proveedores de servicios de impresión entre nuestros clientes por lo que conocemos el flujo de trabajo típico de esta industria. Es importante observar que la mayoría de las impresoras que ofrecen “soporte Web” están limitadas en los archivos que pueden subir y no incluyen realmente un elemento dinámico para realizar pruebas.

Proceso típico en la industria de hoy

El cliente carga sus recursos de impresión que podrían ser cualquier combinación de imágenes, de gráficos, logos y datos de la forma. La impresora crea una prueba que puede ser enviada por fax, correo (típicamente enviado por servicio express) o el cliente puede ir a ver la prueba. Todas estas opciones tienen desventajas obvias.

El mejor ejemplo, entrega por servicio express es costoso y el envío de fax es de baja calidad y no aceptable. La prueba es además estática, es decir no se pueden ver diversos datos de diversos clientes ni las diferentes imágenes que conlleva cada tipo de cliente.

Al final lo importante es como presentar a un cliente con una prueba idéntica al documento impreso y que pueda mostrar un muestreo de datos variables.

Mejora del proceso

Usando la Web más allá de subir archivos provee al cliente un proceso donde pueden ver sus pruebas en línea como PDFs exactamente como se imprimirían.

Usando Opus WebService, el cliente puede cargar sus recursos, seleccionar una plantilla del documento deseado, y ver una prueba en tiempo real incluyendo color y datos variables.

Beneficios

- *Los informes se pueden generar sin la programación en COBOL*
- *Los estudiantes pueden ver sus resultados inmediatamente*
- *Los costos de envío se redujeron significativamente*

Elixir