

Aprovisionamiento de dispositivos móviles

Mike Calligaro

Me llamo Mike Calligaro y llevo 14 años trabajando como desarrollador en Microsoft. Aparte de un breve período de dos semanas en el grupo de escritorio de Windows®, he pasado toda mi carrera en lo que ahora es la división de Windows Mobile. He trabajado con una amplia gama de productos, incluyendo una versión temprana de la televisión interactiva, la videoconsola Sega Dreamcast y el sistema operativo Windows CE. Los últimos ocho años los he dedicado a trabajar en distintas secciones de Windows Mobile.

Hoy hablaré acerca de la configuración mediante programación de dispositivos Windows Mobile (o "aprovisionamiento", para los que les gustan las palabras rimbombantes). Si trabaja como administrador de TI o profesional del soporte técnico y está configurando manualmente dispositivos de Windows Mobile para sus usuarios, déjelo, este artículo va dirigido a usted. Y si se dedica a desarrollar aplicaciones móviles y desea que su página web forme parte de los favoritos del explorador, también debería leer este artículo.

¿Qué pensaría si le dijera que con unas pocas líneas de código y algo de XML usted puede configurar una VPN, establecer la fecha y la hora del dispositivo, configurar una cuenta de correo electrónico, agregar un archivo al dispositivo, establecer un tono de llamada, cambiar una clave del Registro y muchísimas cosas más? ¿Cuánto pagaría por eso? ¿Un millón de euros? ¿Dos millones de euros? Pues bien, gracias a la oferta especial de MSDN® Magazine, podrá conseguirlo por el módico precio de 0 euros. Sí, ha entendido bien, es gratis. Así que llame ahora mismo y no deje pasar esta oportunidad...

Bueno, espero ser mejor desarrollador que vendedor. Poniéndonos serios, las API de aprovisionamiento forman parte del kit de desarrollo de software (SDK) de Windows Mobile, que puede descargarse gratuitamente del Centro de desarrollo de Windows Mobile en msdn.microsoft.com/windowsmobile.

¿Por qué hablamos de aprovisionamiento?

A pesar de que el nombre de Windows Mobile fue algo posterior, Microsoft inauguró su primera plataforma para teléfonos móviles en 2002. La entrada en el sector de la telefonía supuso la implementación de muchas de las cosas que los

operadores de móvil deseaban hacer pero a las que no estaban acostumbrados, como la capacidad de configurar dispositivos de forma remota. Y aunque no había mucho que configurar en los teléfonos del año 2002, necesitábamos apoyar ese nivel básico de configuración para ser competitivos. Una vez organizada la infraestructura, creímos que era conveniente esforzarnos en la configuración remota.

Por ejemplo, debido a mi trabajo, usaba varios teléfonos y diferentes versiones de sistemas operativos. Durante los momentos más complicados del ciclo de desarrollo, en ocasiones he tenido que configurar más de un teléfono varias veces al día. Incluso cuando las cosas van como la seda, no es raro que tenga que configurar al menos un teléfono a la semana. Así que en lugar de escribir la dirección de mi servidor Exchange y establecer mi tono de llamada, formulé un BLOB XML que realiza la mayor parte de la configuración. Luego guardo el archivo XML y un pequeño programa para procesarlo en una tarjeta de memoria y así estoy siempre preparado para cuando alguien me entregue un nuevo teléfono.

Estoy seguro de que tener Penny Arcade como favorito en el explorador nunca ha sido un requisito para los operadores de móvil, pero me alegro de que en este caso nuestra mentalidad haya sido la de "si funciona para una cosa, funcionará para todas".

Para que esta configuración haga maravillas, todo lo que necesita es una sola API (DMProcessConfigXML) y un poco de XML. Por ejemplo, digamos que quiere que Live Search sea un favorito en el explorador. Empiece con el siguiente XML:

```
// WAP provisioning XML to add a new browser favorite.
LPCWSTR g_wszFavoriteXml =
    L"<wap-provisioningdoc> "
    L"  <characteristic type=\"BrowserFavorite\"> "
    L"    <characteristic type=\"Live Search\"> "
    L"      <parm name=\"URL\" value=\"http://m.live.com/\"/> "
    L"    </characteristic> "
    L"  </characteristic>"
    L"</wap-provisioningdoc>";
```

Una vez hecho esto, el único código que necesita para agregarlo como favorito usando XML es el siguiente:

```
HRESULT AddFavorite() {
    HRESULT hr      = E_FAIL;
    LPWSTR  wszOutput = NULL;

    // Process the XML.
    hr = DMProcessConfigXML(g_wszFavoriteXml,
        CFGFLAG_PROCESS, &wszOutput);

    // The caller must delete the XML returned from DMProcessConfigXML.
```

```
delete [] wszOutput;
return hr;
}
```

Lo hago de manera simplificada por razones de espacio. Debería comprobar la seguridad y los errores en su propio código. Es más probable que esté leyendo el XML de un archivo a que lo haya compilado en su código. Pero en resumidas cuentas, eso es lo que necesita hacer.

También hay un contenedor administrado para DMProcessConfigXML. En C# puede usar ConfigurationManager.ProcessConfiguration en su lugar. Éste es el mismo código en C#:

```
XmlDocument configDoc = new XmlDocument();
configDoc.LoadXml(
    "<wap-provisioningdoc>"+
    "<characteristic type=\"BrowserFavorite\">"+
    "<characteristic type=\"Live Search\">"+
    "<parm name=\"URL\" value=\"http://m.live.com/\"/>"+
    "</characteristic>"+
    "</characteristic>"+
    "</wap-provisioningdoc>"
);
ConfigurationManager.ProcessConfiguration(configDoc, false);
```

Proveedores de servicios de configuración

DMProcessConfigXML envía el XML recibido a varios Proveedores de servicios de cifrado (CSP, Configuration Service Providers). Estos proveedores leen el XML y actúan sobre él. Si hay algún tipo de configuración que desee cambiar y que no sea admitida, es probable que todavía no hayamos escrito un CSP para ella. Si cree que hay nuevos CSP que deberíamos agregar, por favor, infórmenos.

La lista definitiva de CSP está disponible en msdn2.microsoft.com/bb737536.aspx. También encontrará varios ejemplos de XML en msdn2.microsoft.com/bb737572.aspx.

Desgraciadamente, todavía existe una gran diferencia de disponibilidad de CSP para los teléfonos Windows Mobile Professional (Pocket PC) y los Windows Mobile Standard (Smartphone). Verá que algunos de los CSP están disponibles para el Standard pero no para el Professional (no, la ironía no nos pasa desapercibida), pero estamos trabajando para que en el futuro las plataformas estén unificadas. Además, los operadores de móviles y los OEM tienen la capacidad de elegir y deshabilitar los CSP para bloquear la funcionalidad que no desean que se modifique. Para estar seguro, debería probar siempre las configuraciones que quiera implementar en los teléfonos.

Y ahora pasemos a los ejemplos. Empezaremos con un BLOB de XML que establece la zona horaria, activa la alarma y la configura:

```
<wap-provisioningdoc>
  <characteristic type="clock">
    <parm name="TimeZone" value="4"/>
    <parm name="AlarmOn" value="1"/>
    <parm name="AlarmTime" value="06:30:00Z"/>
  </characteristic>
</wap-provisioningdoc>
```

El tipo de características le dice que usa el Clock CSP. Cada CSP tiene una página MSDN que describe los parámetros que puede cambiar así como los valores posibles que puede asignar a cada parámetro. Por ejemplo, la documentación para el Clock CSP está disponible en msdn2.microsoft.com/bb737261.aspx. Esta documentación describe los valores posibles para TimeZone, AlarmOn y AlarmTime, así como dos parámetros que no se incluyen en este ejemplo (la fecha y la hora).

No se olvide de mirar en los documentos los valores correctos para asignar a los parámetros. Seguro que para algunos tienen sentido, pero para los simples mortales como yo, no. Por ejemplo, el valor de TimeZone para GMT -8 es 4, pero el valor para GMT -5 es 35.

La [Figura 1](#) muestra un ejemplo de algunas de las cosas que puede hacer con el Sync CSP. La documentación para Sync CSP está en msdn2.microsoft.com/bb737700.aspx. Como verá, puede elegir los datos que desea sincronizar, desde dónde hacerlo, con qué frecuencia, etc. Hay como unos 50 parámetros relacionados con la sincronización que puede configurar con este proveedor.

Aunque estos ejemplos sólo sean una pequeña muestra, esperamos que le den una idea de cómo funcionan los CSP.

Implementación del XML en el dispositivo

Ya ha creado un XML excelente y ha escrito una pequeña aplicación que abre un archivo XML y envía los datos a DMProcessConfigXML. Pero no logrará ningún aprovisionamiento hasta que consiga implantar el XML y la aplicación en el dispositivo.

Para ello tiene varias opciones. La más sencilla consiste en guardar el archivo y la aplicación en una tarjeta de memoria, insertar la tarjeta en el dispositivo y, a continuación, ejecutar la aplicación usando el Explorador de archivos. Esto es lo que yo hago para configurar mis dispositivos.

¿Pero qué ocurre si es administrador de TI y tiene que configurar centenares de dispositivos? Los pasos adicionales necesarios para iniciar el Explorador de archivos y navegar hasta la aplicación nos haría perder mucho tiempo. En ese caso, podría hacer que la aplicación de configuración se ejecutase automáticamente al insertar la tarjeta de memoria.

Cree una carpeta llamada 2577 (no pregunte por qué) en la raíz de la tarjeta de memoria y guarde la aplicación de configuración en esa carpeta. Denomine a la aplicación autorun.exe. Si la aplicación abre automáticamente un archivo con un nombre XML estándar (por ejemplo, config.xml), entonces todo lo que tiene que hacer para configurar el dispositivo es insertar la tarjeta. El truco de AutoRun funciona también cuando el dispositivo arranca, así que funcionará incluso si su teléfono requiere extraer la batería para insertar la tarjeta de memoria.

Por supuesto, hay más opciones. Si después de la instalación inicial desea modificar nuevamente la configuración de sus dispositivos, puede poner la aplicación de configuración en el dispositivo y asociarle una extensión. A continuación, envíe por correo electrónico los archivos de configuración con la extensión asociada a sus usuarios y éstos sólo tendrán que abrir el archivo de configuración adjunto para configurar sus dispositivos. Otra posibilidad sería que los usuarios visitaran un sitio web que realizara la descarga del archivo de configuración por ellos.

O también usar PushRouter para enviar por SMS los archivos de configuración de protocolo WAP a los dispositivos. En este artículo no nos ocuparemos de PushRouter, pero cabe mencionar que es el método que usan los operadores de móvil para enviar archivos de configuración a los teléfonos de sus redes.

Si usted es un fabricante de software independiente (ISV, Independent Software Vendor), puede usar los CSP desde dentro de su programa de instalación para realizar cualquier configuración del dispositivo que sea necesaria, desde agregar archivos para cambiar claves de Registro, hasta agregar favoritos al explorador o cambiar la pantalla principal. La sencillez del formato de configuración también facilita la posibilidad de tener un archivo XML coincidente para que el desinstalador deshaga el trabajo del instalador.

Si dispone de una conexión al dispositivo a través de ActiveSync® o del Centro de Dispositivos de Windows Mobile, podrá configurarlo sin tener que introducir en él ningún archivo. La herramienta de escritorio rapiconfig.exe se encuentra en el directorio de herramientas del kit de desarrollo de software de Windows Mobile. Esta herramienta leerá un archivo XML y usará RAPI para configurar el dispositivo que en ese momento esté conectado a ActiveSync. Después de acoplar el dispositivo, entre en el directorio de herramientas del SDK y (asumiendo que el nombre de su archivo XML es config.xml) escriba:

rapiconfig config.xml

Para obtener más información acerca de RapiConfig, consulte msdn2.microsoft.com/bb737541.aspx.

Finalmente, puede crear un archivo CAB o un archivo de formato de aprovisionamiento de CAB (CPF, CAB Provisioning Format). Tanto los CAB como los CPF se pueden copiar en un dispositivo y ejecutarlos directamente sin necesidad de un programa. La herramienta de escritorio makecab.exe se usa para crear ambos tipos de archivo y también está disponible en el kit de desarrollo de software de Windows Mobile.

La principal diferencia entre los archivos CAB y CPF es que este último no abre ningún cuadro de diálogo cuando se ejecuta (ejecución silenciosa), lo cual constituye también su mayor inconveniente. Dependiendo de los ajustes de seguridad del dispositivo y de si el archivo está firmado o no, los archivos CAB y CPF pueden necesitar un cuadro de diálogo que pregunte al usuario si desea aprovisionar el dispositivo. Puesto que el CPF se ejecuta silenciosamente, no puede mostrar este cuadro de diálogo y, por lo tanto, se produce un error. Para acabar, si está pensando en firmar el archivo de configuración, use un CPF. Si no, use un CAB.

Es muy sencillo crear cualquiera de los dos. Sólo tiene que pasar un archivo XML normal a makecab y darle el nombre de archivo de salida que desee. El truco consiste en que el archivo XML debe llamarse `_setup.xml`. Si usa otra denominación, makecab funcionará correctamente y generará el archivo CAB o CPF, pero se producirá un error críptico cuando intente usarlo. Pero cuidado, la documentación no lo aclara muy bien.

La sintaxis para crear un archivo CAB es la siguiente:

```
makecab _setup.xml config.cab
```

A diferencia del archivo XML, no existen restricciones a la hora de nombrar el archivo CAB generado. Por ejemplo en vez de `config.cab`, usted puede elegir cualquier nombre que desee para el archivo CAB generado. La sintaxis para crear un archivo CPF es la siguiente:

```
makecab _setup.xml config.cpf
```

De nuevo, sustituya `config.cpf` por cualquier otro nombre que desee para el archivo CPF. Sí, correcto, la única diferencia entre los dos es la extensión `.cab` o `.cpf`. Y al igual que antes, en ambos casos el archivo XML debe llamarse `_setup.xml`.

Vaya a msdn2.microsoft.com/bb737689.aspx para obtener más información acerca de los archivos CPF y a msdn2.microsoft.com/bb416436.aspx para hacer lo propio con los archivos CPF.

¿Quién quiere escribir XML?

Escribir XML manualmente está muy bien para los grandes profesionales de la informática, pero, ¿y si quiere permitir a sus usuarios que elijan su propia configuración entre un conjunto de opciones? ¿No sería estupendo tener un sitio web con un grupo de casillas? Los usuarios explorarían el sitio, elegirían lo que quisieran y se generaría el archivo XML.

Para ello, Microsoft® .NET Framework ofrece la clase XmlDocument, facilitando la generación de XML mediante programación. ¿Recuerda, por ejemplo, el clock (reloj) XML que mencionamos anteriormente? El código de la [Figura 2](#) lo crea al momento.

CreateRoot, CreateCharacteristic y CreateParm son funciones de ayuda muy fáciles de conseguir. Luego cree su propio CreateXML para llamar a estas funciones. Este ejemplo codifica la información del parámetro, pero en su página web los valores vendrían de las elecciones de la interfaz del usuario. Después de generar el XML, el sitio web lo descarga en el dispositivo para que lo ejecute el usuario.

También es posible escribir una aplicación de configuración que se ejecute en el dispositivo. Después de que el usuario seleccione la configuración apropiada, un código como el de la [Figura 2](#) generaría el XML y lo enviaría directamente a ConfigurationManager.ProcessConfiguration. Así, usted se ahorraría el proceso necesario para enviar el XML a un archivo y volver a leerlo más tarde.

Y ya está todo listo y aprovisionado.

¿Alguna vez ha tenido que entrar en una dirección de servidor Exchange usando el teclado numérico de su teléfono? Vuelva a la comodidad de su teclado de escritorio y redacte un poco de XML; nadie merece pasar por semejante trago. Espero que este artículo le haya parecido... liberador. Feliz aprovisionamiento.

Envíe sus preguntas y comentarios a goplaces@microsoft.com.

Mike Calligaro es Jefe de Desarrollo de Windows Mobile de Microsoft y colaborador del blog de Windows Mobile en blogs.msdn.com/windowsmobile.

© 2007 Microsoft Corporation and CMP Media, LLC. Reservados todos los derechos; queda prohibida la reproducción parcial o total sin previa autorización.