

IBM Endpoint Manager
Versión 9.2

Guía de acciones



IBM Endpoint Manager
Versión 9.2

Guía de acciones



Nota

Antes de utilizar esta información y el producto al que da soporte, lea la información en "Avisos" en la página 73.

Esta edición se aplica a la versión 9, release 2, nivel de modificación 0 de IBM Endpoint Manager y a todos los releases y modificaciones posteriores mientras no se indique lo contrario en nuevas ediciones.

© Copyright IBM Corporation 2010, 2015.

Contenido

Capítulo 1. Introducción al lenguaje de acciones 1

Creación de scripts de acción	1
Introducción del bloque de captación previa	2
Utilización de la sustitución	3
Introducción a las descargas dinámicas	4
Descarga estática	4
Descarga dinámica	4

Capítulo 2. Mandatos de ejecución 9

action launch preference low-priority	9
action launch preference normal-priority	9
dos	10
notify client ForceRefresh	10
override	11
Completion	12
Priority (solo Windows)	12
Hidden (solo windows)	13
Detached (solo Windows).	13
RunAs	13
run	13
rundetached	14
runhidden	14
script	15
wait	16
waitdetached	16
waithidden	17

Capítulo 3. Mandatos de control de flujo 19

action may require restart	19
action parameter query	19
action requires login	20
action requires restart	20
continue if	21
exit	21
if, elseif, else, endif	22
Captación previa	23
parámetro	24
pause while	25
restart	26
set clock	26
shutdown	26

Capítulo 4. Mandatos del sistema de archivos 29

action log	29
add nohash prefetch item	29
add prefetch item	30
appendfile	31
archive now	32
begin prefetch block	33
collect prefetch items	34
copy	35

createfile until	36
suprimir	37
download	37
download as	39
download now as	40
end prefetch block	41
execute prefetch plug-in	42
extract	43
folder create	43
folder delete	44
move	45
prefetch	45
relay select	46
utility	47

Capítulo 5. Establecimiento de mandatos 49

establecer	49
setting delete	50

Capítulo 6. Mandatos de registro 51

regdelete	51
regset	51

Capítulo 7. Mandatos Wow64 55

action uses wow64 redirection	55
regdelete64	55
regset64	56
script64	57

Capítulo 8. Mandatos de derechos administrativos 59

administrator add	59
administrator delete	59

Capítulo 9. Mandatos de mantenimiento de cliente de BigFix 61

module add	61
module commit	61
module delete	62

Capítulo 10. Bloqueo de mandatos. 63

action lock indefinite	63
action lock until	63
action unlock	63

Capítulo 11. Mandatos de mantenimiento del sitio 65

site force evaluation	65
site gather schedule disable	65
site gather schedule manual	65
site gather schedule publisher	66
site gather schedule seconds	66

subscribe 66
darse de baja 67

Términos y condiciones de la documentación del
producto 76

Capítulo 12. Comentarios. 69
barra inclinada doble 69

Apéndice. Asistencia 71

Avisos 73
Marcas registradas 75

Capítulo 1. Introducción al lenguaje de acciones

Después de que un Fixlet identifique un problema potencial en un sistema, ofrece arreglarlo con un mandato de shell de IBM Endpoint Manager, denominado script de acción. Aunque hay otras maneras de crear scripts, el método más potente consiste en utilizar el **Lenguaje de acciones de IBM Endpoint Manager**, porque se integra estrechamente con el motor de relevancia.

Muchos mandatos de acción permiten o requieren parámetros. Estos parámetros pueden ser valores o expresiones codificados (estáticos) que el motor de relevancia de IBM Endpoint Manager evalúa e inserta. Estos se denominan **variables de sustitución** y le permiten crear scripts con objetivos bien marcados y extremadamente flexibles. La expresión de relevancia exacta que haya desencadenado la acción se puede utilizar en el script de acción, garantizando la corrección del problema. Todos los mandatos pueden realizar sustitución en sus argumentos antes de procesarlos, con sólo unas excepciones señaladas.

En este documento se describen todos los mandatos de acción de IBM Endpoint Manager, con ejemplos específicos. En la parte inferior de cada tema de acción aparece un número de versión, por ejemplo, Versión 7.2 y posteriores. Esto representa la primera versión que es compatible con el mandato indicado. Algunas acciones se marcan con "Solo Windows", y no se ejecutarán correctamente en sistemas UNIX o Macintosh.

Creación de scripts de acción

Puede crear acciones personalizadas para solucionar o abordar problemas en la red que no quedan cubiertos por el contenido estándar. Aunque el proceso se explica fácilmente, hay una amplia variedad de acciones y técnicas de marcación de objetivos a su disposición. Para crear una acción personalizada:

1. Inicie la sesión en la consola de IBM Endpoint Manager como operador maestro.
2. Seleccione **Herramientas > Llevar a cabo una acción personalizada**.
3. Se abre el diálogo **Llevar a cabo una acción**. En la parte superior hay una zona para poner **Nombre** a la acción personalizada. Este campo se puede ordenar y filtrar, por tanto, es conveniente utilizar una convención adecuada para los nombres a fin de sacar el máximo provecho de los informes.
4. Bajo el campo Nombre, el menú desplegable **Preconfiguración** permite elegir una acción personalizada preconfigurada, lo que le permite ahorrar tiempo y garantizar su precisión. También puede guardar su trabajo actual como preconfiguración para volver a utilizarlo posteriormente. La interfaz de preconfiguración incluye estos campos y botones:
 - a. **Preconfigurado**: Seleccione una preconfiguración en el menú desplegable.
 - b. **Mostrar sólo las personales**: Marque esta casilla para filtrar la lista de preconfiguraciones y que sólo muestre las suyas.
 - c. **Guardar preconfiguración**: Guarde el conjunto de opciones de acción actual para su uso posterior. Este botón no estará activo hasta que haga algún cambio en alguna de las opciones de este diálogo. Cuando pulse este botón, se abrirá un diálogo que le solicitará el nombre de la preconfiguración. Debajo verá una casilla de verificación que permite guardar la preconfiguración como pública o privada.

- d. **Eliminar preconfiguración:** Elimina esta preconfiguración de la lista seleccionable. Se abre un diálogo de confirmación que le permite cancelar este mandato.
5. Bajo el área de las preconfiguraciones, hay varias fichas:
 - a. **Destino:** Seleccione los objetivos en la lista proporcionada, o bien utilice las propiedades o una lista específica de equipos para marcar los objetivos de la acción.
 - b. **Ejecución:** Especifique las opciones y restricciones de implementación, incluida la recuperación tras error y aplicaciones repetidas.
 - c. **Usuarios:** Determine cómo responderá esta acción a la presencia o ausencia de usuarios.
 - d. **Mensajes:** Proporcione un mensaje que precederá y acompañará a la acción.
 - e. **Oferta:** Cree una oferta de acción, que permite al usuario seleccionar si desea o no aplicar la acción.
 - f. **Post-acción:** Describa qué acciones deben llevarse a cabo para completar la acción, incluyendo el reinicio o el apagado.
 - g. **Aplicabilidad:** Permite alterar la relevancia de acción original.
 - h. **Criterios de éxito:** Cree criterios específicos que puede utilizar para determinar si la acción ha tenido éxito.
 - i. **Script de acción:** Esta ficha le permite crear o modificar un script de acción.
 6. Pulse la ficha **Script de acción** y escriba su script. Esta guía describe los mandatos de acción disponibles y proporciona varios ejemplos.
 1. Pulse en la ficha **Aplicabilidad** si desea ajustar los objetivos del script de acción. Para obtener más información sobre el lenguaje de relevancia, consulte la *Referencia del lenguaje de relevancia de IBM Endpoint Manager* y las *Guías de inspectores de IBM Endpoint Manager*.
 2. Pulse las fichas **Ejecución**, **Usuarios**, **Mensajes**, **Oferta** o **Post-acción** para personalizar más la acción.
 3. Cuando esté preparado para implementar la acción personalizada, pulse **Aceptar**.
 4. Su acción personalizada se distribuirá a todos los equipos que se hayan seleccionado o marcado como objetivos. Las acciones se aplicarán con las restricciones y planificaciones que haya especificado.

También puede crear acciones cuando realice las acciones **Crear tareas** o **Crear Fixlets**. Consulte el manual de la Consola de IBM Endpoint Manager para obtener más información sobre estos temas.

Introducción del bloque de captación previa

El bloque de captación previa debe ser la primera entrada del script de acción (aparte de comentarios o líneas en blanco). Contiene toda la lógica de captación previa de descarga necesaria para prepararse para la posterior ejecución de la acción, facilitando la comprensión de la acción. Algunos de los métodos que se pueden utilizar en un bloque de captación previa incluyen:

Descargas literales

Estas son descargas estáticas normales, que siguen estando disponibles.

Descargas condicionales

Sólo se ejecutan mandatos dentro de las rutas de condición TRUE.

Sustitución de variables

Incluye las descargas que utilizan la sustitución de relevancia para determinar qué archivos recopilar.

Lógica personalizada

Aprovecha un plugin para crear manifiestos de descarga.

A diferencia del algoritmo de preanálisis que se utiliza en las acciones de descarga tradicionales, las descargas de bloque de captación previa se pueden visualizar como un enfoque descendente: el bloque de captación previa va primero y debe completarse satisfactoriamente antes de que el resto de la acción pueda continuar. Esto proporciona un mayor control, flexibilidad y potencia.

Nota: Solo se permite un bloque de captación previa por acción. Si se utiliza, el mandato **begin prefetch block** debe ser el primer ejecutable del script. Sólo puede ir precedido de líneas en blanco y comentarios. Se requiere un mandato **end prefetch block** para terminar.

Utilización de la sustitución

La sustitución permite al autor del Fixlet incluir expresiones de relevancia en una acción. Este paso se lleva a cabo colocando la expresión de relevancia entre llaves: `run "{pathname of regapp "excel.exe"}"`

Este ejemplo ejecuta un programa sin saber dónde se encuentra. Una expresión de relevancia evalúa el nombre de la ruta automáticamente utilizando el inspector 'regapp'.

```
pause while {exists running application "c:\updater.exe"}
```

Esta acción se detiene hasta que un programa termine de ejecutarse, utilizando el inspector `running application`.

La sustitución no es recursiva, aunque es posible que algún mandato determinado tenga una o más expresiones para evaluar antes de la ejecución. El cliente de IBM Endpoint Manager espera encontrar una única expresión dentro de los corchetes. Si ve otro corchete izquierdo antes de que detecte un corchete derecho de cierre, lo tratará como un carácter normal:

```
echo {"a left brace: {"}
```

podría enviar esta cadena de caracteres para salida:

```
a left b race: {
```

Por lo tanto, no se necesitan caracteres de escape especiales para representar un corchete izquierdo. Para extraer un corchete derecho literal sin terminar la sustitución, utilice un carácter doble:

```
echo {"{a string inside braces}"}
```

podría enviar esta cadena de caracteres para salida:

```
{a string inside braces}
```

O considere este ejemplo:

```
appendfile {{ nombre del sistema operativo} {nombre del sistema operativo}
```

Cuando este ejemplo se analiza, los corchetes izquierdos dobles indican que lo que sigue a continuación no es una expresión de relevancia. Solo se necesita un único

corchete derecho cuando está fuera de una expresión de relevancia (dentro de una expresión de relevancia, se necesita un corchete derecho para especificar uno literal). Esto enviará la siguiente línea a `__appendfile`:

```
{ nombre del sistema operativo } WinXP
```

También puede utilizar la sustitución con los mandatos **add prefetch item** en bloques de captación previa:

```
begin prefetch block
  parameter "manifest"="{pathname of file "manifest.spec" of client folder
    of site "AV"}"
  add prefetch item {concatenation " ; " of lines of file
    (parameter "manifest")}
end prefetch block
```

Introducción a las descargas dinámicas

La característica de descarga dinámica amplía la flexibilidad de los scripts de acción. Para comprender cómo funciona, resulta útil conocer el método de descarga estática existente.

Descarga estática

Antes de iniciar una acción, el cliente de IBM Endpoint Manager la analiza, buscando mandatos `download` o `prefetch`. Las descargas estáticas incluyen el URL, el algoritmo de hash SHA y el tamaño de cada elemento como valores literales en el script de acción. Los valores literales permiten a un operador observar qué va a hacer exactamente el script de acción. Estos valores literales se utilizan para construir una lista numerada de descargas asociadas a la acción que se almacena en el servidor de IBM Endpoint Manager. Esta etapa del proceso de la acción se denomina **proceso de captación previa**.

Como consecuencia del proceso de captación previa, el cliente notificará al retransmisor de IBM Endpoint Manager más cercano sobre las descargas necesarias solicitando un URL que termina con el `<id_acción>/0`, que a su vez desencadena el retransmisor para descargar todos los elementos correspondientes a esa acción especificada. Cuando están preparadas, el retransmisor hace ping a los clientes con el ID de acción. Todos los clientes de IBM Endpoint Manager que ejecuten esa acción recopilarán los archivos, solicitándolos de uno en uno como `<id_acción>/1`, `<id_acción>/2`, etc.

Sin embargo, debido a que la información de descarga se representa mediante expresiones literales, sólo se podrán representar los URL ya conocidos cuando se creó la acción. Esto significa que las descargas estáticas no pueden utilizarse para las instancias en las que las descargas cambian, pero el script de acción permanece igual.

Descarga dinámica

Las descargas dinámicas añaden la capacidad de utilizar cláusulas de relevancia para especificar las descargas. Estos nuevos mandatos deben incorporarse en un segmento especial de código de acción denominado **bloque de captación previa**. Por ejemplo, si ha creado un archivo en el sitio de AV Fixlet denominado `download.spec`, que contiene una variable designada en la primera línea como:

```
name=update.exe sha1=123 sha256=678 size=456 url=http://site.com/download/patch.exe
```

Podrá acceder a este parche utilizando la sustitución de relevancia en un bloque de captación previa:

```
begin prefetch block
  parameter "downloadFile"="{pathname of file "download.spec" of client folder
    of site "AV"}"
  add prefetch item {line 1 of file (parameter "downloadFile")}
end prefetch block
```

Este bloque de código crea una variable denominada `downloadFile` que apunta a un archivo en el sitio de AV. Luego añade este archivo a la cola de captación previa para su descarga posterior. De esta forma, un mensaje de Fixlet en el sitio de AV podría ofrecer mantener algo actualizado automáticamente y el archivo `download.spec` se renovaría cada vez que hubiera disponible una nueva versión. La implementación de la acción desde el Fixlet como una acción de política ejecutaría la actualización siempre que el archivo `download.spec` se hubiera modificado.

Tenga en cuenta que este bloque de código finaliza con un mandato **end prefetch block**, lo cual garantizará que el archivo se descargue correctamente antes de ejecutar el script de acción. Un bloque de captación previa debe estar en la parte superior del script de acción, y debe cerrarse con la sentencia `end prefetch block` antes de que el script pueda continuar.

Otra técnica muy utilizada consiste en utilizar un archivo de datos, o manifiesto, que contenga una lista de varias descargas, cada una con su propio URL, algoritmo de hash SHA y tamaño. Este manifiesto puede cambiar tan a menudo como sea necesario, simplificando la actualización de definiciones de spy ware o antivirus. Una forma de implementar esta acción es crear un archivo denominado `manifest.spec` con una lista de descargas como, por ejemplo:

```
name=patch1.exe sha1=123 sha256=347 size=456 url=http://site.com/download/patch1.exe
name=patch2.exe sha1=234 sha256=358 size=567 url=http://site.com/download/patch2.exe
name=patch3.exe sha1=345 sha256=368 size=678 url=http://site.com/download/patch3.exe
A continuación, puede descargar estos parches con un bloque de captación
previa que extrae estos archivos del manifiesto:
begin prefetch block
  parameter "manifest"="{pathname of file "manifest.spec" of client folder
    of site "AV"}"
  add prefetch item {concatenation " ; " of lines of file
    (parameter "manifest")}
end prefetch block
```

También puede utilizar pequeños ejecutables para procesar los archivos en un manifiesto nuevo. Este paso se lleva a cabo con el mandato **execute prefetch plug-in**, como ilustra el ejemplo siguiente:

```
begin prefetch block
  add prefetch item name=myPlugIn.exe sha1=123 size=456
    url=http://mysite/plugin.exe sha2=347
  // collect the plug-in before continuing:
  collect prefetch items
  parameter "ini"="{file "prepass.ini" of site (value of setting
    "CustomSite") of client}"
  execute prefetch plug-in "{download path "myPlugIn.exe"}" /downloads
    "{parameter "ini"}" "{download path "manifest"}"
  add prefetch item {concatenation " ; " of lines of download file
    "manifest"}
end prefetch block
```

Este bloque de captación previa primero añade el plugin a la cola de captación previa y, a continuación, ejecuta el mandato **collect prefetch items**. Esto hace que el proceso de captación previa se retrase hasta que los elementos añadidos a la cola

de captación previa se descarguen antes de continuar el proceso de captación previa. Una vez descargado correctamente, el plugin se ejecuta con argumentos, incluida la ruta para el archivo de datos y el manifiesto que se producirá a partir del mismo. El mandato **add prefetch item** final pone en cola las descargas especificadas en el manifiesto recién creado. Una técnica como esta también puede utilizarse para descifrar un archivo seguro en un manifiesto sin formato.

Las descargas dinámicas deben especificar archivos con la confirmación de un tamaño o algoritmo de hash SHA. El URL, el tamaño y el algoritmo de hash SHA pueden provenir de un origen fuera del script de acción. Esta flexibilidad implica un nivel adicional de escrutinio. Ya que cualquier cliente puede utilizar las descargas dinámicas para solicitar un archivo, supone una oportunidad para que otras personas puedan utilizar su servidor para alojar archivos de forma indiscriminada. Para evitarlo, las descargas dinámicas usan una lista blanca. Cualquier petición de descarga desde un URL (que no esté autorizada explícitamente mediante el uso de un URL literal en el script de acción) debe cumplir uno de los criterios especificados en una lista blanca de URL en el servidor de IBM Endpoint Manager, ubicado en <ruta de instalación del servidor de BES>\Mirror Server\Config\DownloadWhitelist.txt. Este archivo contiene una lista de expresiones regulares separadas por saltos de línea que emplea el formato de Perl, como en los siguientes ejemplos:

```
http://.*\.sitename\.com/.  
http://software\.sitename\.com/.  
http://download\.sitename\.com/patches/JustThisOneFile\.qfx
```

La primera línea es la menos restrictiva, permite descargar cualquier archivo del dominio del nombre del sitio. La segunda línea requiere un host específico de dominio y la tercera es la más restrictiva, ya que limita el URL a un único archivo denominado "JustThisOneFile.qfx". Si un URL solicitado no cumple con ninguna entrada de la lista blanca, la descarga falla inmediatamente con el estado **No disponible**.

En el registro del retransmisor se crea una nota que indica la dirección URL que no se ha permitido.

Una lista blanca vacía o no existente provocará que las descargas dinámicas fallen. Una entrada de lista blanca de ".*" (punto asterisco) permitirá la descarga de cualquier URL.

Los bloques de captación previa permiten las sentencias condicionales:

```
begin prefetch block  
    if {name of operating system = "Windows 2000"}  
add prefetch item name=up.exe sha1=123 size=456  
    url=http://site.com/patch2k.exe sha2=567  
    else  
add prefetch item name=up.exe sha1=123 size=456  
    url=http://site.com/patch.exe sha2=567  
    endif  
end prefetch block  
wait "{download path "up.exe"}"
```

Este script de acción se ramifica en la existencia de Win2K, pero las descargas de este ejemplo se describen estáticamente (como texto literal). Aunque los clientes sólo descargarán determinados elementos que necesiten, todos los archivos estáticos se descargan a los servidores y retransmisores tan pronto como se solicitan. Las descargas dinámicas pueden mejorar esta situación porque sólo se captan en el servidor y retransmisor los archivos que realmente necesiten los clientes. Aquí se muestra un ejemplo de utilización de las descargas dinámicas:

```
begin prefetch block  
    if {name of operating system = "Windows 2000"}  
add prefetch item {"name=up.exe sha1=123 size=456  
    url=http://site.com/patch2k.exe"} sha2=567  
    else
```

```

add prefetch item {"name=up.exe sha1=123 size=456
    url=http://site.com/patch.exe"} sha2=567
    endif
end prefetch block
wait "{download path "up.exe"}"

```

Utilizando la sustitución de relevancia en el bloque de captación previa, con un archivo de lista blanca configurado correctamente en el servidor, este código sólo capta el archivo necesario, mejorando potencialmente los requisitos de ancho de banda y la eficacia.

También puede ramificar la ejecución en base al contenido de un archivo, lo que le permite automatizar las actualizaciones. Este pueden ser especialmente útil para gestionar el cambio de números de versión. Por ejemplo, podría crear un archivo denominado 'manifest.txt' que contenga dos variables designadas como:

```

version=1234
download=name=update.exe sha1=123 size=456
    url=http://site.com/download/patch.exe sha2=567

```

Tenga en cuenta que la variable de descarga contiene el nombre, sha1, sha2, tamaño y URL del archivo de parche.

Puede utilizar la sustitución de relevancia para extraer estas variables con una expresión como:

```

parameter "ver"="{key "version" of file "{download path "manifest.txt"}"}"
parameter "filename"="{key "download" of file "{download path "manifest.txt"}"}"

```

Si compara la versión extraída con algunos valores almacenados, puede determinar si necesita, y cuándo, descargar el archivo especificado. Esta técnica se puede ampliar para incluir varias versiones y puede incluso utilizarse para distinguir entre parches y actualizaciones de sustitución completa.

Independientemente de la técnica utilizada, una vez que se hayan descargado los archivos, pueden examinarse con varios inspectores, que pueden tener distintas interpretaciones, en función de si la acción está activa (o en el proceso de captación previa). Antes de la ejecución, estos archivos se recopilan en una carpeta de captación previa. Durante la ejecución de la acción, residen en la carpeta `__Download`.

Hay nuevos inspectores que se pueden utilizar para localizar los archivos antes o durante la ejecución de la acción:

- **carpeta de descargas:** Durante el análisis de captación previa, este inspector devuelve un objeto de carpeta desde la carpeta `__Global\<nombreSitio>\<idAcción>\named`. Cuando la acción está activa y la descarga se ha completado, este inspector devuelve el objeto de carpeta esperado desde el directorio `__Download`.
- **Ruta de descarga "pathname":** Este inspector devuelve una cadena que contiene el nombre de ruta completo hasta el archivo especificado, si existe o no. El nombre de archivo de descarga es equivalente a (nombre de ruta de la carpeta de descarga) & <separadorRuta> & nombreArchivo.
- **Archivo de descarga "filename":** Este inspector devuelve un objeto de archivo desde la carpeta de descargas u otra carpeta designada. El nombre de archivo de descarga es equivalente a "archivo 'filename' de la carpeta de descarga". Si el archivo no está todavía en la carpeta de descargas, el inspector devuelve 'does not exist'.

Depende del autor del script de acción el proteger a los usuarios de estas acciones y garantizar que las descargas y sus sumas de comprobación no se revelen. Un mecanismo de autenticación completo, resistente a ataques de hombre en medio (man-in-the-middle), es la mejor defensa. Al crear una acción de descarga dinámica resulta fundamental crear la acción de manera que autentique la información antes

de utilizarla, normalmente utilizando un plugin, tal como se describe anteriormente. También es recomendable identificar explícitamente esos pasos en el script de acción que lleva a cabo esta autenticación, de manera que los usuarios de la acción puedan auditar el mecanismo antes de decidir si confían en él.

Nota: Solo se permite un bloque de captación previa por acción. Si se utiliza, el mandato **begin prefetch block** debe ser el primer mandato del script. Sólo puede ir precedido de líneas en blanco y comentarios. Se requiere un mandato **end prefetch block** para separar los bloques de captación previa del resto de la acción.

Capítulo 2. Mandatos de ejecución

En esta sección se describen los mandatos de ejecución.

action launch preference low-priority

Cuando se ejecuta este mandato, los mandatos de acción subsiguientes que inicien programas lo harán con prioridad inferior al valor normal. Esto le ayudará a mitigar el impacto de grandes parches o actualizaciones de Service Pack.

La preferencia de prioridad baja sólo afecta a la prioridad de lanzamiento de aplicaciones iniciadas desde la acción actual. Esta preferencia se mantiene hasta que la acción se complete o hasta que el cliente ejecute el mandato **action launch preference normal-priority**.

Sintaxis

```
action launch preference low-priority
```

Ejemplos

```
action launch preference low-priority
run "{pathname of regapp "background_app.exe"}"
action launch preference normal-priority
```

Este ejemplo reduce la prioridad de lanzamiento antes de ejecutar `background_app` para que no domine el sistema cuando se ejecute. A continuación, establece el nivel de prioridad de nuevo al valor normal.

th

Nota: Este mandato es solo de Windows. Hará que un script de acción finalice en un agente UNIX.

Versión 6.0 y posteriores -- solo Windows

action launch preference normal-priority

Cuando se ejecuta este mandato, los mandatos de acción subsiguientes que inicien programas lo harán con prioridad normal. Esta sentencia sólo es necesaria para restablecer la prioridad normal después de un mandato **action launch preference low-priority**.

Sintaxis

```
action launch preference normal-priority
```

Ejemplos

```
action launch preference low-priority
run "{pathname of regapp "background_app.exe"}"
action launch preference normal-priority
```

Este ejemplo reduce la prioridad de lanzamiento antes de ejecutar `background_app`, luego restablece la prioridad normal para las sentencias de lanzamiento posteriores.

Notas

Este mandato es solo de Windows. Hará que un script de acción termine en un agente Unix.

Versión 6.0 y posteriores -- solo Windows

dos

Emite un mandato de DOS estándar. Si el mandato de DOS falla, el script de acción que lo contiene finaliza.

Sintaxis

dos <línea de mandatos de DOS>

Ejemplo

```
dos rmdir /Q /S "{pathname of windows folder & "\temp"}"
```

Este ejemplo suprime un directorio vacío desde una carpeta temporal del directorio de ventanas.

```
dos scandisk.exe e:
```

En este ejemplo, `e:` es un parámetro que se ha pasado al programa `scandisk`.

Notas

Este mandato es solo de Windows. Hará que un script de acción termine en un agente Unix.

En un sistema Windows, tiene el mismo efecto que la emisión de una sentencia de sistema (sintaxis de línea de mandatos de DOS) desde la API de Windows. Es lo mismo que escribir la línea de mandatos de DOS en una solicitud de DOS. El mandato de DOS utiliza la variable de entorno `PATH` para intentar ubicar el mandato en la unidad de disco duro del usuario. Igual que con cualquier mandato de DOS, para otras ubicaciones debe especificar un nombre de ruta completo.

Asegúrese de utilizar comillas si tiene espacios en los nombres de archivo.

Versión 5.1 y posteriores -- solo Windows

notify client ForceRefresh

Este mandato es equivalente a pulsar con el botón derecho del ratón en un equipo cliente en la consola de IBM Endpoint Manager y seleccionar **Enviar actualización**. Este mandato puede ser necesario si la conexión UDP al cliente de IBM Endpoint Manager Client está bloqueada.

Sintaxis

override

El mandato `override` proporciona la capacidad de personalizar determinados mandatos y añadir varias variaciones a los mandatos existentes. Este potente mandato compuesto le permite crear su propio mandato de combinación personalizado similar a los mandatos existentes `waitdetached` o `runhidden`. Para añadir restricciones a un mandato existente, se añaden pares palabra clave/valor predefinidos en el cuerpo del mandato.

Sintaxis

```
override <cmd>  
<keyword>=<valor>  
<keyword>=<valor>  
<cmd> <resto de la línea de mandatos>
```

Donde **cmd** es el mandato `wait` o `run`, y los pares palabra clave/valor se seleccionan de una tabla mostrada en la sección Notas de uso.

Ejemplo

```
override wait  
hidden=true  
wait notepad.exe
```

Este ejemplo proporciona la misma funcionalidad que `waithidden notepad.exe`.

```
override wait  
completion=job  
hidden=true  
runas=currentuser  
wait __Download\patch.exe arg1 arg2 arg3
```

Este ejemplo muestra cómo puede ejecutar un parche como un proceso oculto por el usuario actual, esperando la finalización del trabajo antes de continuar con el script de acción.

Notas de uso

Las palabras clave se pueden especificar en cualquier orden, pero debe haber solo uno por línea. No es necesario dejar espacios en blanco alrededor de '=' (signo igual) y se ignora.

Las palabras clave no distinguen entre mayúsculas y minúsculas, y los valores pueden ir entre {llaves} para la sustitución de relevancia. Si se listan palabras clave duplicadas, se utilizará el último valor. El mandato no se ejecutará si alguna de las palabras clave o valores no son válidos. Las palabras clave específicas de la plataforma que no tienen sentido en una plataforma determinada se ignorarán de forma silenciosa. Como con la mayoría de los mandatos, la cadena de mandato se ejecuta a través de la evaluación de la relevancia, de modo que la comprobación de la sintaxis del lado de la consola es mínima. Sin embargo, el agente realiza la comprobación completa durante la ejecución de la acción.

Las palabras clave disponibles y sus valores se muestran en la tabla siguiente.

palabra clave	valores permitidos	valor predeterminado
completion	none process job	none process *
priority	normal low	normal
hidden	false true	false
detached	false true	false
runas	agent currentuser	agente

* para run, el valor predeterminado es none, para wait, el valor predeterminado es process.

Completion

- Completion=**none** actúa del mismo modo que las variantes del mandato run actuales.
- Completion=**process** actúa del mismo modo que las variantes del mandato wait actuales.
- Completion=**job** en Windows utiliza el JobObject de Windows que impone algunas limitaciones en el proceso de destino y algunos posibles puntos de error para el mandato. Consulte los detalles a continuación.

Limitaciones de 'completion=job'

Windows:

Para ejercer el control de trabajos más flexible durante un proceso, el mandato de alteración temporal permite el proceso de separar de forma selectiva procesos hijo del trabajo. Esto permite que el proceso realice su propia gestión de control de trabajos, pero elimina cualquiera de sus hijos separados del objeto de trabajo.

En dichos casos limitados, en los que el proceso iniciado es responsable de su propio control de trabajos, se supone que un miembro del trabajo permanecerá en ejecución hasta que todos sus procesos hijo se hayan completado. Sin embargo, esto no garantiza nada y es posible que haya situaciones en las que no sea este el caso. En tales casos, la acción se completa aunque los procesos hijo todavía estén ejecutándose.

UNIX/Linux:

En las plataformas UNIX/Linux, los ID de sesión se utilizan para gestionar procesos de trabajo. Los ID de sesión toman el valor del id de proceso del líder de la sesión (el proceso que desea iniciar). El cliente espera a que finalice el proceso principal, como en el caso 'completion=process', a continuación, inicia un ciclo de medio segundo de suspensión seguido de la enumeración de los procesos buscando cualquier dispositivo con un id de sesión que coincida con el ID de proceso del líder de trabajo. Cuando ya no hay más de estos procesos, el trabajo se completa y el mandato finaliza.

El código de salida devuelto con el mandato siempre es el del líder de proceso, no del último proceso en completarse.

Versión 8.2 y posteriores

Priority (solo Windows)

- Priority=**normal** actúa del mismo modo que el mandato 'action launch priority normal' actual.

- Priority=**low** actúa del mismo modo que el mandato 'action launch priority low' actual.

Hidden (solo windows)

- Hidden=**true** aplica el atributo SW_HIDE al proceso al igual que con los mandatos runhidden y waithidden.
- Hidden=**false** elimina el atributo SW_HIDE del proceso.

Detached (solo Windows)

- Detached=**true** crea el proceso utilizando el método de separación como se realiza en los mandatos rundetached y waitdetached.
- Detached=**false** crea el proceso mediante el método normal

RunAs

- RunAs=agent aplica las mismas características de propiedad de proceso que las variantes wait y run actuales.
- RunAs=currentuser imita el RunAsCurrentUser.exe actual en Windows, utilizando la misma lógica para identificar el usuario actual y código similar para crear el proceso con un bloque de entorno originado por userToken. En UNIX/Linux, no puede obtener universalmente las variables de entorno de usuario adecuadas, por lo que no se intenta aplicar en absoluto las variables de entorno, con la excepción de las variables Xauthority necesarias. En UNIX/Linux se realiza una llamada a setuid para el ID del usuario identificado como el usuario actual del XBESClientUI. Se trata de una prueba muy específica y dependiente de la plataforma, que requiere que el usuario inicie sesión en la consola local y se ejecute en X Windows.

Para obtener información adicional, consulte la sección "override" en la página 11.

run

Ejecuta el programa indicado. Si no se puede crear el proceso, el script de acción finaliza. El mandato run no espera que el proceso finalice antes de ejecutar la siguiente línea del script de acción. La línea de mandatos contiene el nombre del ejecutable y opcionalmente puede contener parámetros. Si desea esperar a que un programa finalice antes de iniciar otro, utilice el **mandato wait**.

Sintaxis

run <línea de mandatos>

Ejemplos

```
run "{pathname of regapp "wordpad.exe"}"
run "c:\winnt\ftp.exe" ftp.mycorp.net
run wscript /e:vbs x.vbs arg1 arg2
```

Estos ejemplos muestran cómo se debe ejecutar un script y pasarle algunos argumentos. Se recomienda el uso de comillas en la línea de mandatos, siendo necesarias si hay espacios en los nombres de archivo.

Nota

En un sistema Windows, este mandato tiene el mismo efecto que llamar a la API `CreateProcess` con la <línea de mandatos>. Esto es lo mismo que utilizar la <línea de mandatos> en un diálogo RUN de Windows. Consulte la documentación de Windows sobre `CreateProcess` para obtener más información sobre el método que se utiliza para buscar el ejecutable desde una <línea de mandatos>.

Versión 5.1 y posteriores

rundetached

`Rundetached` modifica el mandato **run** estableciendo el indicador `DETACHED_PROCESS` al llamar a `CreateProcess()` en máquinas Windows. De forma predeterminada, un proceso creado hereda su consola padre. Cuando se desconecta, este comportamiento se inhibe. Esta opción ofrece al nuevo proceso más control sobre cómo puede interactuar con el usuario.

Entre otras cosas, se puede utilizar para evitar ventanas de DOS emergentes al ejecutar un programa. Es lo mismo que el mandato **run**, pero el proceso creado no accede a la consola padre, lo que impide la molesta ventana de DOS. `Rundetached` no debe utilizarse para ejecutar programas interactivos. Si se lleva a cabo, el programa interactivo no podrá mostrar su interfaz de usuario y puede parecer que se haya colgado. Este mandato se proporciona estrictamente para ejecutar programas que no muestran una interfaz de usuario.

Sintaxis

rundetached <línea de mandatos>

Ejemplos

```
rundetached "{pathname of regapp "background_app.exe"}"  
rundetached "c:\winnt\ftp.exe" ftp.filesite.net
```

Estos ejemplos muestran cómo se debe ejecutar un programa y pasarle algunos argumentos. Se recomienda el uso de comillas en la línea de mandatos, siendo necesarias si hay espacios en los nombres de archivo.

Notas

Este mandato es solo de Windows. Hará que un script de acción termine en un agente Unix. En un sistema Windows, tiene el mismo efecto que la emisión de una sentencia `CreateProcess(CommandLine)` desde la API de Windows. Esto es lo mismo que utilizar `CommandLine` en un diálogo RUN de Windows. Consulte la documentación de Windows sobre `CreateProcess()` para obtener más información sobre el método que se utiliza para buscar el ejecutable desde una `CommandLine`.

Versión 5.1 y posteriores -- solo Windows

runhidden

Este mandato utiliza `CreateProcess()` para iniciar un mandato en una ventana oculta. Oculta la ventana estableciendo `STARTUPINFO dwFlag` en `STARTF_USESHOWWINDOW` y estableciendo `wShowWindow` en `SW_HIDE`. El proceso que se crea puede modificar el distintivo para posteriormente volver a mostrar la ventana.

Después de iniciarse, se ejecuta la siguiente línea de mandato de acción. Para esperar que se complete el lanzamiento antes de continuar la acción, utilice el mandato **waithidden**.

Sintaxis

runhidden <línea de mandatos>

Ejemplos

```
runhidden "{pathname of regapp "wordpad.exe"}"  
runhidden "c:\winnt\ftp.exe" ftp.mycorp.net  
runhidden wscript /e:vbs x.vbs arg1 arg2
```

Estos ejemplos muestran cómo se debe ejecutar un script en una ventana oculta y pasarle algunos argumentos. Se recomienda el uso de comillas en la línea de mandatos, siendo necesarias si hay espacios en los nombres de archivo.

Notas

Este mandato es solo de Windows. Hará que un script de acción termine en un agente Unix.

Si el proceso iniciado requiere entrada del usuario, la esperará con su ventana oculta, a menos que el mandato muestre su ventana explícitamente.

En un sistema Windows, este mandato tiene el mismo efecto que llamar a la API `CreateProcess()` con la <línea de mandatos> y establecer los indicadores para ocultar la ventana. Consulte la documentación de Windows sobre `CreateProcess()` para obtener más información sobre el método que se utiliza para buscar el ejecutable desde una <línea de mandatos>.

Versión 6.0 y posteriores -- solo Windows

script

No debe confundirse con un script de acción, la palabra clave del script ejecuta un script externo (creado para un lenguaje de scripts como JavaScript o Visual Basic) con el nombre proporcionado. El script de acción que contiene la palabra clave terminará si no está instalado el motor de scripts adecuado o si el script no se puede ejecutar. La línea siguiente de la acción Shell Command no se ejecuta hasta que el script especificado finaliza.

Sintaxis

script <nombre de script>

Ejemplo

```
script attrib.vbs
```

Ejecuta el script de Visual BASIC `attrib.vbs`.

Notas

Este mandato es solo de Windows. Hará que un script de acción termine en un agente Unix.

En un sistema Windows, tiene el mismo efecto que la emisión de una sentencia `wscript "scriptName"` desde Windows, y la posterior espera hasta que se completa. Esto es lo mismo que utilizar `scriptName` desde un diálogo RUN de Windows. Si necesita pasar parámetros al script, utilice el mandato **run** en su lugar.

Versión 5.1 y posteriores -- solo Windows

wait

El mandato `wait` se comporta igual que el mandato **run**, excepto que espera a que el programa o el proceso finalice antes de continuar.

Sintaxis

wait <línea de mandatos>

Ejemplo

```
wait "scandskw.exe"
```

Ejecuta el programa `scandskw` y espera a que el programa finalice antes de continuar con el script de acción. Se recomienda la utilización de comillas, siendo necesarias si hay espacios en los nombres de archivo.

Nota

En un sistema Windows, tiene el mismo efecto que la emisión de una sentencia `CreateProcess <línea de mandatos>` desde la API de Windows, y la posterior espera hasta que se completa.

Versión 5.1 y posteriores

waitdetached

`Waitdetached` se utiliza para evitar ventanas de DOS emergentes cuando se espera que un programa se complete. Es lo mismo que el mandato **wait**, pero el proceso creado no accede a la consola padre, lo que impide la molesta ventana de DOS. `Rundetached` no debe utilizarse para ejecutar programas interactivos. Si se lleva a cabo, el programa interactivo no podrá mostrar su interfaz de usuario y puede parecer que se haya colgado. Este mandato se proporciona estrictamente para ejecutar programas que no muestran una interfaz de usuario.

Sintaxis

waitdetached <línea de mandatos>

Ejemplo

```
waitdetached "scandskw.exe"  
waitdetached wscript /e:vbs x.vbs arg1 arg2
```

Este ejemplo muestra cómo puede ejecutar un script, pasarle algunos argumentos y luego esperar a que se complete antes de continuar el script de acción.

Notas

Este mandato es solo de Windows. Hará que un script de acción termine en un agente Unix.

En un sistema Windows, tiene el mismo efecto que la emisión de una sentencia CreateProcess (CommandLine) desde la API de Windows, y la posterior espera hasta que se completa.

Versión 5.1 y posteriores -- solo Windows

waithidden

Este mandato es similar al mandato **runhidden** y utiliza CreateProcess para ejecutar un mandato en una ventana oculta. Oculta la ventana estableciendo STARTUPINFO dwFlag en STARTF_USESHOWWINDOW y estableciendo wShowWindow en SW_HIDE. Esta acción espera a que el proceso se complete antes de continuar con los siguientes mandatos de acción.

Sintaxis

waithidden <línea de mandatos>

Ejemplos

```
waithidden "{pathname of regapp "notepad.exe"}"  
waithidden "c:\winnt\ftp.exe" ftp.myurl.net  
waithidden wscript /e:vbs x.vbs arg1 arg2
```

Estos ejemplos muestran cómo se debe ejecutar un script en una ventana oculta y pasarle algunos argumentos. Se recomienda el uso de comillas en la línea de mandatos, siendo necesarias si hay espacios en los nombres de archivo.

Notas

Este mandato es solo de Windows. Hará que un script de acción termine en un agente Unix.

Si el proceso iniciado requiere entrada del usuario, la esperará con su ventana oculta, a menos que el mandato muestre su ventana explícitamente.

En un sistema Windows, este mandato tiene el mismo efecto que llamar a la API CreateProcess con la <línea de mandatos> y establecer los indicadores para ocultar la ventana. Consulte la documentación de Windows sobre CreateProcess para obtener más información sobre el método que se utiliza para buscar el ejecutable desde una <línea de mandatos>.

Versión 6.0 y posteriores -- solo Windows

Capítulo 3. Mandatos de control de flujo

action may require restart

Cuando se ejecuta este mandato, el cliente busca signos en el sistema que indiquen que es necesario reiniciar. Si es así, establece el estado de finalización de la acción de manera que la acción aparecerá como "Pendiente de reinicio" en la consola, hasta que se reinicie. Cuando el reinicio se ha completado, el estado de finalización de la acción tomará el valor de 'success' si la relevancia de la acción ya no es relevante, o 'failed' si aún es relevante.

Si no existen signos que indiquen que es necesario reiniciar, el estado de finalización de la acción tomará el valor de 'success' si la relevancia de la acción ya no es relevante, o 'failed' si aún es relevante.

Sintaxis

action may require restart

Ejemplo

```
action may require restart
```

Versión 5.1 y posteriores

action parameter query

Este permite que la entrada de datos de parámetros esté disponible a través de la relevancia durante la ejecución de la acción. Los nombres de parámetros pueden incluir espacios en blanco y son sensibles a mayúsculas y minúsculas. El nombre del parámetro, la descripción y el valor deben estar contenidos cada uno dentro de comillas dobles (""). Una vez introducida, la entrada del usuario se convierte en el valor predeterminado en las siguientes invocaciones (para Endpoint Manager, el usuario es el operador de consola que aprueba la acción para el despliegue).

Sintaxis

action parameter query "<nombre de parámetro>" [with description "<descripción>"] [and] [with default [value] "<valor predeterminado>"]

Donde **nombre de parámetro** es el nombre del parámetro de relevancia y la opción **with description** permite presentar una solicitud al usuario. La opción **and with default** le permite especificar un valor predeterminado para el parámetro.

Ejemplos

```
action parameter query "InstallationPoint" with description "Please enter the location of the shared installation point:"
action parameter query "Registry key" with description "Please enter your desired registry key" and with default value "null"
action parameter query "tips" with description "Enter 'on' or 'off' to control Fixlet tips." With default "on" regset "[HKEY_CURRENT_USER\Software\BigFix]"
"tips"="{parameter "tips" of action}"
```

Nota: Los valores de parámetro especificados por el usuario pueden incluir %xx, donde xx significa un número hexadecimal de dos dígitos para definir el carácter que desea incorporar. Para incorporar un signo de porcentaje, utilice %25. Para incorporar una comilla doble, utilice %22.

Mientras se ejecuta la acción, puede recuperar el valor de parámetro de acción especificado por el operador de consola. Por ejemplo, en la acción podría utilizar la sustitución de relevancia: {parameter "parameter name" of action}.

La sustitución de relevancia **NO** se realiza en la propia línea de mandatos **action parameter query**. Esto se debe a que el mandato se interpreta en la consola de IBM Endpoint Manager antes de que la acción se envíe, lo que permite al autor del Fixlet solicitar al operador los parámetros específicos para el despliegue, necesarios para ejecutar la acción.

Versión 5.1 y posteriores

action requires login

Este mandato informa al cliente que la acción actual no se completará hasta que se reinicie el equipo y un administrador inicie sesión. Cuando esta acción se haya completado en una máquina, el inspector **pendiente de inicio de sesión** devolverá true (verdadero).

Sintaxis

action requires login

Ejemplo

```
action requires login
```

Nota

Los agentes de Unix IBM Endpoint Manager ignoran esta acción.

Versión 5.1 y posteriores

action requires restart

Este mandato informa al cliente que la acción actual no se completará hasta que se complete el próximo reinicio. Cuando esta acción se haya completado en una máquina, el inspector 'pendiente de inicio de sesión' devolverá true (verdadero). Si hay un mandato 'action requires restart' en una acción, la consola de IBM Endpoint Manager notificará 'Pendiente de reinicio' hasta que la máquina afectada se reinicie.

Sintaxis

action requires restart

Ejemplo

```
action requires restart
```

Versión 5.1 y posteriores

continue if

Este mandato permite ejecutar la siguiente línea en el script si el valor proporcionado como parámetro se evalúa con el valor true. Se detendrá sin error si el valor especificado se evalúa con el valor false. Puede utilizar la sustitución de relevancia para calcular el valor. Este mandato es útil para garantizar que se cumplan determinadas condiciones para ejecutar el resto de una acción. El número de línea donde ha salido el script de acción se notifica a la consola. Los usuarios de IBM Endpoint Manager pueden utilizar este número de línea para identificar la causa por la cual falla una acción si inserta una sentencia **continue if** que identifica una invariable que la acción necesita.

Sintaxis

```
continue if <condición de true>
```

Donde **condición de true** representa una expresión de relevancia para evaluar.

Ejemplos

```
continue if {name of operating system = "Win2k"}  
download now http://www.real-time.com/downloads/win98/dun40.exe
```

Este ejemplo descargará el archivo dun40.exe sólo si el sistema operativo es Win 2000.

```
continue if {(size of it = 325 and sha1 of it = "013e48a5") of file  
  "dun40.exe" of folder "__Download"}  
wait __Download/dun40.exe /Q:A /R:N
```

Este ejemplo ejecutará el archivo dun40.exe sólo si se especifica el tamaño y el valor sha1.

Versión 5.1 y posteriores

exit

El mandato exit finaliza la acción y devuelve un valor entero al interlocutor.

```
exit {integer exit code}
```

Este mandato puede utilizar la sustitución de relevancia. Cuando se ejecuta, el valor del entero se transfiere al inspector de código de salida, y la acción finaliza en esa línea. Los códigos de salida pueden afectar al estado 'fixed' de las acciones de ejecución hasta el final, si salen del script antes de la última línea. Si no hay líneas ejecutables después del mandato de salida, la acción se completará satisfactoriamente. Sin embargo, si hay otros mandatos después del mandato de salida, la acción de ejecución hasta el final fallará.

En la consola de IBM Endpoint Manager, el valor del último mandato de salida se muestra en el diálogo Ver información de acción, junto con otra información de estado.

En una expresión de relevancia, este valor puede evaluarse utilizando el inspector 'exit code of <acción>'.

Sintaxis

exit <{expresión}>

Donde **expresión** es el valor entero que se devuelve al interlocutor. Esto está limitado a números asignados de 32 bits, sin embargo en Unix, el límite es 8 o 16 bits, que puede determinarse ejecutando la macro WIFEXITED.

Ejemplo

```
wait 'foo'
parameter "error" = "{exit code of action}"
if {parameter "error" != "0"}
    exit {parameter "error"}
endif
// continue processing
```

Este ejemplo representa un script que notifica errores como códigos de salida que no son cero. Esto le permite terminar el script de forma prematura e informar el código de salida al interlocutor. Este es uno de los cuatro mandatos de script (`wait`, `waithidden`, `waitdetached` y `exit`) que pueden modificar el valor de inspector del código de salida. Los tres mandatos `wait` definen el código de salida de acuerdo con el ejecutable, con el sistema operativo limitando el tamaño del número. El mandato `exit` define el código de salida de acuerdo con el número que se le pase como número de 32 bits asignado, independientemente del sistema operativo.

Puede utilizar el mandato `exit` junto con DOS, sin embargo DOS no puede definir el propio código de salida debido a una limitación de la API de mandatos del sistema.

Nota para scripts de shell de Unix: Para las acciones de tipo 'x-sh', el código de salida del script se recopila en el valor de inspector cuando el cliente termina de procesar un script de shell. Los códigos de salida desde scripts de shell de UNIX se graban en el registro de clientes.

Versión 8.0

if, elseif, else, endif

Los mandatos `if`, `elseif`, `else` y `endif` permiten la ejecución condicional de los mandatos de acción. Estas sentencias condicionales trabajan con expresiones entre llaves, como en el siguiente esquema:

```
if {EXPR1}
    statements to execute on EXPR1 = TRUE
elseif {EXPR2}
    statements to execute on EXPR1 != TRUE and EXPR2 = TRUE
else
    statements to execute when EXPR1 != TRUE and EXPR2 != TRUE
endif
```

En el esquema de acción anterior, si la expresión entre llaves seguida de la sentencia `if` es true (verdadera), se evalúan las sentencias siguientes (hasta la sentencia `endif`). Los bloques `If` se pueden anidar a cualquier cantidad de niveles de profundidad.

Se aplica la semántica de bloque `if` normal. Todas las sentencias hasta un `endif`, `elseif` o `else` constituyen un bloque. Las sentencias `elseif {EXPR}` y `else` son opcionales. Puede utilizarse cualquier número de sentencias `elseif`, pero solo un bloque `else` de seguimiento.

Captación previa

El cliente de IBM Endpoint Manager analiza las acciones antes de ejecutarlas, buscando descargas para captar previamente. Si el proceso de captación previa no analiza correctamente, se devuelve un error de **sintaxis de acción** y la acción no se ejecuta. Esto puede causar problemas si está creando acciones que funcionan en varios entornos, donde sólo una rama de una sentencia `if` puede analizar correctamente. Por ejemplo, es posible que desee cargar archivos que son exclusivos de plataformas específicas.

Un script como este funcionaría:

```
if {not exists key "foo" of registry}
    prefetch windows_file ...

else if {not exists package "bar" of rpm}
    prefetch UNIX_file ...

endif
```

Aquí la clave de registro de Windows desencadena la primera captación previa, mientras que un paquete UNIX desencadena la segunda. El problema es que el inspector de registros no se ejecutará correctamente en los sistemas UNIX, y el inspector de paquetes no se ejecutará correctamente en Windows, provocando que el analizador de captación previa genere un error en ambos casos.

La solución consiste en utilizar inspectores de plataforma cruzada (como `name of operating system`) para asegurarse de que no se evalúen los bloques incorrectos:

```
if {name of operating system starts with "Win"}
    if {not exists key "foo" of registry}
        prefetch windows_file ...
    endif

else if {name of operating system starts with "Redhat"}
    if {not exists package "bar" of rpm}
        prefetch UNIX_file ...
    endif

Endif
```

Al comprobar primero el sistema operativo adecuado, puede evitar este tipo de error de análisis de captación previa. Sin embargo, a veces puede que no haya ninguna manera de evitar un posible error. Por ejemplo, una acción puede crear y acceder a un archivo que no existen todavía en la fase de captación previa:

```
wait chkntfs c: > c:\output.txt
if {line 2 of file "c:\output.txt" as lowercase contains "not dirty"}
    regset "HKLM\Software\MyCompany\" "Last NTFS Check"="OK"

else
    regset "HKLM\Software\MyCompany\" "Last NTFS Check"="FAIL"

endif
```

En este ejemplo de Windows, el archivo de salida no existe hasta que el script se ejecuta. El analizador de captación previa advertirá que el archivo no existe cuando se compruebe su contenido. A continuación, generará un error y terminará la acción. Sin embargo, puede ajustar la condición `if` para permitir el paso de captación previa. Una técnica consiste en utilizar la expresión `"not active of action"` que siempre devuelve `TRUE` durante el paso de captación previa. Puede utilizar esto para evitar bloques problemáticos durante el análisis previo:

```

wait chkntfs c: > c:\output.txt
if {not active of action OR (line 2 of file "c:\output.txt" as lowercase
    contains "not dirty")}
    regset "HKLM\Software\MyCompany\" "Last NTFS Check"="OK"

else
    regset "HKLM\Software\MyCompany\" "Last NTFS Check"="FAIL"

endif

```

Al comprobar primero para ver si la acción se está preanalizando o ejecutando, obtendrá un paso de captación previa correcto y el comportamiento deseado cuando se ejecute la acción.

Sintaxis

```

if {<expresión>}
    <sentencias>
endif

```

Ejemplo

```

if {name of operating system = "WinME"}
    prefetch patch1.exe sha1:e6dd60e1e2d4d25354b339ea893f6511581276fd size:4389760
    http://download.microsoft.com/download/whistler/Install/310994/WIN98MeXP/EN-US
    /WinXP_EN_PRO_BF.EXE
    wait __Download\patch1.exe

elseif {name of operating system = "WinXP"}
    prefetch patch2.exe sha1:92c643875dda80022b3ce3f1ad580f62704b754f size:813160
    http://www.download.windowsupdate.com/msdownload/update/v3-19990518/cabpool
    /q307869_f323efa52f460ea1e5f4201b011c071ea5b95110.exe
    wait __Download\patch2.exe

else
    prefetch patch3.exe sha1:c964d4fd345b6e5fd73c2235ec75079b34e9b3d2 size:845416
    http://www.download.windowsupdate.com/msdownload/update/v3-19990518/cabpool
    /q310507_2f3c5854999b7c58272a661d30743abca15caf5c.exe
    wait __Download\patch3.exe

endif

```

Este fragmento de código capta previamente, cambia el nombre y descarga un archivo, en función del sistema operativo.

Versión 6.0 y posteriores

parámetro

El mandato parameter puede utilizarse para crear nuevas variables de acción durante la ejecución de la acción. Tiene el formato:

```
parameter "x" = "{expression}"
```

Este mandato permite acceder al parámetro utilizando el inspector **parameter "x"**. El parámetro sólo se puede inspeccionar dentro de la acción actual. Los parámetros se han inicializado justo antes de que se inicie la acción desde encabezados añadidos a la acción por la consola de IBM Endpoint Manager.

No puede restablecer un parámetro que ya tiene un valor. Cuando esto suceda, el cliente termina anormalmente la acción en la línea que está intentando restablecer el parámetro. Cualquier error resultante de la evaluación de la expresión se realizará convirtiendo el parámetro designado como indefinido.

Las reglas del mandato parameter son:

Las expresiones del parámetro se forzarán en cadenas de caracteres.

Las expresiones plurales que resultan sin valores darán como resultado un valor de parámetro vacío.

Las expresiones plurales que resultan en un único valor que puede forzarse en una cadena asignarán el valor.

Las expresiones plurales que resultan en más de un valor darán como resultado un error de la acción.

Sintaxis

parameter "<x>" = "<{expresión}>"

Donde x es el nombre del parámetro y **expresión** es el valor. Tenga en cuenta que tanto el nombre del parámetro como la expresión deben estar entre comillas.

Ejemplo

```
parameter "loc" = "{pathname of folder (value of variable "tmp" of environment)}"
createfile until end
    Operating system = {name of operating system}
    Processor count = {number of processors}
end
delete "{parameter "loc"}\config.txt"
copy __createfile "{parameter "loc"}\config.txt"
```

Define un parámetro denominado "loc" que contiene el nombre de la ruta de la carpeta tmp, crea un nuevo archivo name=value que contiene el sistema operativo y el recuento de procesador, suprime el archivo de configuración de la carpeta tmp y lo sustituye por el nuevo archivo.

Versión 6.0 y posteriores

pause while

La acción no continuará con el siguiente mandato mientras la expresión de relevancia especificada se evalúa con el valor True. Continuará y ejecutará el siguiente mandato de la acción tan pronto como el valor se evalúe con false o que el valor no se pueda evaluar. Utilice la sintaxis de la sustitución de relevancia para definir la condición.

Sintaxis

pause while <condición de true>

Donde **condición de true** representa una expresión de relevancia para evaluar.

Ejemplos

```
pause while {exists running application "updater.exe"}
pause while {not exists file "C:\70sp3\result.log"}
pause while {not exists section "ResponseResult" of file "C:\70sp3\result.log"}
```

Versión 5.1 y posteriores

restart

El mandato restart reiniciará el equipo. Si se proporciona el parámetro opcional <segundos de retraso>, el apagado se realizará automáticamente transcurrido el retraso especificado.

Si un usuario ha iniciado la sesión, se visualizará un diálogo que muestra la cuenta atrás del retraso. En este caso, la interfaz tendrá un botón **Reiniciar ahora** en lugar de un botón **Cancelar**. Asimismo, cuando se muestra la interfaz de usuario del cliente, existe un retraso mínimo de 60 segundos antes de reiniciar.

Si el parámetro de retraso no se ha especificado, se le solicitará que pulse un botón para reiniciar el sistema.

Sintaxis

restart [<segundos de retraso>]

Donde **segundos de retraso** es un parámetro opcional para proporcionar un tiempo de retardo antes de reiniciar.

Ejemplo

```
restart 180
```

Reinicia el equipo en tres minutos.

Nota

El reinicio retardado es un reinicio forzado, no solicitará al usuario que guarde los cambios realizados en documentos, etc. El equipo se reiniciará sin más solicitudes.

Versión 5.1 y posteriores

set clock

Hace que el cliente se vuelva a registrar en el servidor de registro, y que establezca su reloj a la hora recibida desde el servidor durante la interacción. Resulta útil cuando el reloj del cliente no está sincronizado. Este mandato solo de BES no está disponible cuando el cliente opera bajo una licencia de evaluación.

Sintaxis

set clock

Ejemplo

```
set clock
```

Versión 5.1 y posteriores

shutdown

El mandato shutdown es similar al mandato **restart**, pero simplemente apaga el equipo y no lo reinicia.

Si se proporciona el parámetro opcional <segundos de retraso>, el apagado se realizará automáticamente transcurrido el retraso especificado.

Si un usuario ha iniciado la sesión, se visualizará una interfaz de usuario que muestra la cuenta atrás del retraso. En este caso, la interfaz tendrá un botón **Apagar ahora** en lugar de un botón **Cancelar**.

Si el parámetro de retraso no se ha especificado, se le solicitará que pulse un botón para apagar el sistema.

Sintaxis

shutdown [<segundos de retraso>]

Donde **segundos de retraso** es un parámetro opcional para proporcionar un tiempo de retardo antes de apagarse.

Ejemplo

```
shutdown 180
```

Este mandato apaga el equipo en tres minutos.

AVISO

El apagado retardado es una conclusión forzada; no solicitará al usuario que guarde los cambios realizados en documentos, etc. El equipo se apagará sin más solicitudes.

Versión 5.1 y posteriores

Capítulo 4. Mandatos del sistema de archivos

action log

El mandato `action log` le permite especificar cómo se guardarán los registros de acciones. En general, todos los aspectos de una acción, incluidos los mandatos y los parámetros, se registran. Sin embargo, los parámetros pueden contener información sobre el establecimiento de claves privadas o el descifrado de contraseñas. Si desea mantener estas acciones privadas, utilice el mandato `action log` con un tipo de mandato. De esta manera se impide el registro de parámetros potencialmente confidenciales.

Sintaxis

`action log <tipo>`

Donde **tipo** es uno de:

`command`

`all`

Ejemplo

```
action log all
```

Registra los mandatos y los parámetros de las acciones.

```
action log command
```

Registra sólo el mandato ejecutado por la acción, y ninguno de los parámetros.

Versión 8.2 y posteriores

add nohash prefetch item

Los clientes y los retransmisores recopilan archivos por ID de acción y un número ordinal que identifican el archivo exacto que se utilizará, en lugar de cualquier archivo proporcionado por el URL. Los algoritmos de hash SHA1 o SHA256 *no están permitidos* con este mandato de captación previa.

Este mandato añade un elemento de descarga a la cola de captación previa. Debe residir entre un mandato **begin prefetch block** y **end prefetch block**. Se trata de un mandato singular y sólo puede especificar una única descarga a la vez. La sustitución de relevancia no está permitida con los argumentos de este mandato, lo que permite a IBM Endpoint Manager Server copiar en caché la descarga cuando se crea la acción. Si el cliente solicita algún archivo ordinal, el retransmisor los recopilará todos. El cliente sólo descarga el elemento si este mandato está en un bloque de condición TRUE.

Sintaxis

```
add nohash prefetch item [name=<n>] [size=<s>] url=<u>
```

Donde:

- <n> Es un nombre opcional, limitado a 32 caracteres (incluidos caracteres alfanuméricos, guiones, subrayados y puntos no iniciales). Si no se especifica explícitamente ningún nombre, el nombre se extraerá del componente final del URL (después de la barra inclinada final). Sólo se puede especificar un elemento de descarga por mandato.
- <s> Es un tamaño de archivo opcional. Aunque no es obligatorio, si se sabe y se especifica, el programa puede proporcionar información significativa sobre el progreso.
- <u> Es un url obligatorio. Si el nombre no se especifica, se extraerá del componente final del URL proporcionado.

Los argumentos pueden estar en cualquier orden que desee, pero los mandatos no reconocidos generan un error de sintaxis.

Ejemplo

```
begin prefetch block
  add nohash prefetch item url=http://www.mysite/downloads/download25.exe
end prefetch block
wait {download path "download25.exe"}
```

En este ejemplo se utiliza una descarga estática en un bloque de captación previa y la recupera sin un hash. Esta técnica es insegura intrínsecamente, pero utiliza una lista blanca en IBM Endpoint Manager Server para validar el URL.

Versión 7.2 y posteriores

add prefetch item

Añade un elemento de descarga a la cola de captación previa. Este mandato debe residir entre un mandato **begin prefetch block** y **end prefetch block**. Este mandato puede especificar varias descargas separadas por punto y coma.

El servidor de IBM Endpoint Manager almacena en caché la descarga cuando se crea la acción, a menos que se emplee la sustitución de relevancia. Los retransmisores de IBM Endpoint Manager solo recopilan los archivos que los clientes solicitan y los clientes solo solicitarán un archivo si el mandato está dentro de un bloque de condición TRUE.

En lugar de listar los elementos de descarga en la línea de mandatos, puede colocarlos en un archivo (un elemento por línea) y, a continuación, utilizar una sustitución de relevancia como la siguiente:

```
{concatenation ";" of lines of file <su archivo>}
```

Se trata de un uso común cuando se especifica un archivo en un sitio de Fixlet que contiene la información de descarga.

```
add prefetch item [name=<n>] sha1=<h1> sha256=<h2> size=<s> url=<u> [; ...]
```

Donde:

- <n> Es un nombre opcional, limitado a caracteres alfanuméricos, guiones, subrayados y puntos no iniciales. Si no se especifica explícitamente ningún nombre, el nombre se extraerá del componente final del URL (después de la barra inclinada final).

- <h1> Es el sha1 necesario para el archivo especificado.
- <h2> Es el sha256 necesario para el archivo especificado.
- <s> Es el tamaño de archivo necesario.
- <u> Es un url obligatorio. Si el nombre no se especifica, se extraerá del componente final del URL proporcionado.
- [; ...] Indica que el mandato es plural; se pueden especificar archivos adicionales, cada uno de ellos separado por un punto y coma.

Se permite la sustitución de relevancia con los argumentos de este mandato; sin embargo cuando se utiliza de sustitución, el servidor de IBM Endpoint Manager no puede almacenar en caché el elemento de descarga en el momento de creación de la acción.

Los argumentos puede estar en cualquier orden que desee, y los argumentos no reconocidos se ignorarán.

Cuando se utiliza la sustitución de relevancia *without*, los clientes y retransmisores de IBM Endpoint Manager recopilan estos archivos por ID de acción y ordinales. Cuando se utiliza la sustitución de relevancia *with*, los clientes y retransmisores recopilarán estos archivos por URL y algoritmo de hash SHA. Para especificar una descarga sin especificar el algoritmo de hash SHA, utilice el mandato **add nohash prefetch item**.

Ejemplo

```
begin prefetch block
  if {name of operating system = "Windows 2000"}
    add prefetch item {"name=up.exe sha1=12 size=45 url=http://ms.com
                      /hot2k.exe"}
  else
    add prefetch item {"name=up.exe sha1=12 size=45 url=http://ms.com
                      /hot.exe"}
  endif
end prefetch block
wait {download path "up.exe"}
```

Este ejemplo muestra una descarga condicional en un bloque de captación previa. Al comprobar primero el sistema operativo, solo se realizará la captación previa al archivo adecuado, lo que le permitirá ahorrar tiempo y ancho de banda.

Versión 7.2 y posteriores

appendfile

El mandato appendfile crea un archivo de texto denominado **__appendfile** en el directorio del sitio (por defecto C:\Program Files\BigFix__Data\<<nombre del sitio>). Cada vez que invoca el mandato, este añade el texto especificado al final del archivo. Este mandato puede ser útil para crear los archivos de diagnóstico o para crear dinámicamente los archivos que incorporan los atributos de la máquina del usuario final. Este archivo se suprime automáticamente cuando se inicia la acción Shell Commands.

Sintaxis

appendfile <texto>

Donde **texto** representa la información que se va a colocar en el archivo.

Ejemplos

```
appendfile This file will contain details about your computer
appendfile Operating System={name of operating system}
appendfile Windows is installed on the {location of windows folder} drive
```

Los mandatos anteriores registran el sistema operativo y la ubicación de Windows en el archivo `append`

```
appendfile {"Disk " & name of it & ", free space=" & free space of it
as string) of drives}
```

El ejemplo anterior registra el nombre y el espacio libre disponible para todas las unidades en el PC cliente.

Nota

Utilice el mandato **appendfile** como parte de una acción que crea un script que posteriormente se pasa a un intérprete de scripts. Por ejemplo, puede utilizar la siguiente sintaxis para crear un archivo `.ini` utilizando mandatos de acción:

```
appendfile [HKR]
appendfile HostBasedModemData\Parameters\Driver,ModemOn,1,00,00
delete {location of system folder}\smcfg.ini
copy __appendfile {location of system folder}\smcfg.ini
run smcfg
```

Esta misma técnica se puede utilizar para crear archivos `.bat`, archivos `.cmd`, scripts de visual basic, scripts de bash shell, etc.

Versión 5.1 y posteriores

archive now

Este mandato invoca al Archive Manager. Si la modalidad de operación de archivado se establece en manual, este mandato desencadenará el archivado y la carga del conjunto de archivos configurado. Para establecer la modalidad de archivado adecuada en manual, utilice este valor:

```
_BESClient_ArchiveManager_OperatingMode = 2
```

El mandato **archive now** devolverá un estado de Failed si la modalidad de operación no se ha establecido en manual. También devolverá Failed si un archivo existente se está cargando actualmente.

Sintaxis

archive now

Ejemplos

```
archive now
```

Este mandato inicia el archivar y la carga del conjunto de archivos configurado.

Versión 5.1 y posteriores

begin prefetch block

Inicia un conjunto de mandatos para descargar archivos. Normalmente, cuando se descarga un archivo utilizando acciones de IBM Endpoint Manager, la suma de comprobación se evalúa para asegurarse de su autenticidad. Sin embargo, si el destino de la descarga está en flujo, por ejemplo, una definición de antivirus, este requisito puede ser demasiado restrictivo. Para manejar situaciones como esta, IBM Endpoint Manager proporciona mandatos de descarga dinámica, que se especifican entre corchetes mediante **begin prefetch block** y **end prefetch block**. Para obtener más información, consulte *Introducción del bloque de captación previa* e *Introducción a descargas dinámicas*.

Esta función se integra plenamente con su estructura de retransmisor de IBM Endpoint Manager, optimizando las velocidades de descarga y el ancho de banda. Cuando una acción solicita un archivo, el retransmisor comprueba su memoria caché, e inmediatamente reenvía el archivo si está disponible. De lo contrario, la solicitud pasa la línea hasta que alcanza el servidor de IBM Endpoint Manager.

La descarga dinámica utiliza una lista blanca para garantizar que sólo se acceda a sitios de confianza. La lista blanca está contenida en el archivo siguiente:

En los sistemas de Windows:

```
<ruta de instalación del servidor de BES>\Mirror Server\Config\  
DownloadWhitelist.txt
```

En los sistemas Linux:

```
<ruta de instalación del servidor de BES>/Mirror Server/config/  
DownloadWhitelist.txt
```

Este archivo contiene una lista de URL con un formato de expresiones regulares como, por ejemplo, `http://.*\mysite\.com/.*`. El URL que proporcione en una sentencia de captación previa debe coincidir con una entrada en la lista blanca antes de se pueda descargar. Si el URL no se encuentra en la lista, el mandato falla con un error **No disponible**.

Estos mandatos existentes se permiten dentro del bloque de captación previa:

```
// líneas de comentarios y líneas en blanco  
if/elseif/else/endif  
parameter  
action parameter query (tratado como un comentario por el cliente)
```

Los siguientes mandatos nuevos están permitidos dentro del bloque de captación previa, pero no están permitidos fuera del mismo:

```
add prefetch item  
add nohash prefetch item  
collect prefetch items  
execute prefetch plug-in
```

Sintaxis

begin prefetch block

Solo se puede utilizar un bloque de mandatos de captación previa en un script de acción y debe cerrarse con un mandato **end prefetch block**.

Solo comentarios o líneas en blanco pueden preceder a este mandato. Al procesar acciones con bloques de captación previa, **download**, **download as** y **prefetch no**

están permitidos en ninguna parte del script de acción. El mandato **download now as** está permitido, pero *no dentro o antes del bloque de captación previa*.

Ejemplo

```
// script de acción para actualizar automáticamente un manifiesto de URL desde
// un sitio personalizado
begin prefetch block
  parameter "ini"="{file "server_bf.ini" of site (value of setting
    "MyCustomSite") of client}"
  // captación previa del plugin que proporciona la lista de descargas
  add prefetch item name=plugin.exe sha1=123 sha256=347 size=12
    url=http://www.mysite/downloads/myplugin.exe
  // recopilar el archivo de captación previa anterior (necesario para crear un
  // manifiesto compuesto por URL)
  collect prefetch items
  // ejecutar el plugin que genera un manifiesto desde el archivo de datos ini
  execute prefetch plug-in "{download path "plugin.exe"}" /downloads
    "{parameter "ini"}"
    "{download path "urllist"}"
  // Manifiesto de URL con un formato de líneas que contienen: name=<n> sha256=<h2> size=<s>
    url=<url>
  add prefetch item {concatenation " ; " of lines of download file "urllist"}
end prefetch block
// la acción ahora está activo, actualizar los archivos:
waithidden "{download path "plugin.exe"}" /update "{parameter "ini"}" "{location of
  download folder}"
```

Este ejemplo descarga un plugin que procesa otro archivo para generar un manifiesto con una lista de más archivos para descargar. Cuando el bloque de captación previa finaliza, los archivos se descargan y se mueven a la carpeta de descargas, y el resto de la acción puede continuar.

Notas:

Las consolas y los clientes anteriores rechazarán los scripts de acción que utilizan la nueva funcionalidad de captación previa y los identificarán como que contienen errores de sintaxis. Los retransmisores anteriores no procesarán acciones de descarga dinámica, incluso si el servidor y los clientes pueden manejarlas. Solo se permite un bloque de captación previa en un script de acción.

Determinados mandatos no deben aparecer en ningún lugar de un script de acción que contenga un bloque de captación previa: **download**, **download as** y **prefetch**. **Download now** puede aparecer en el script, pero debe ir *después* del bloque de captación previa.

Se han añadido diversos inspectores nuevos para permitir que el script de acción haga referencia a los archivos descargados mediante la sustitución de relevancia. Se incluyen **download path** "*<nombre>*", **download file** "*<nombre>*" y **download folder**. Consulte la Guía de inspectores para obtener más información.

Versión 7.2 y posteriores

collect prefetch items

Después de que los mandatos hayan añadido los elementos a la cola de captación, como por ejemplo **add nohash prefetch items** y **add prefetch items**, este mandato recopila los elementos del retransmisor de IBM Endpoint Manager. El proceso de captación previa de la acción se suspende hasta que se recopilan todos los archivos

especificados. Si el mandato **add prefetch** ha proporcionado las descargas con nombres nuevos, en este punto se les cambiará el nombre.

Este mandato se utiliza normalmente para recuperar un plugin y/o un conjunto de archivos que puede ser procesado por un plug-in. En este caso, primero se añade un archivo a la lista de captación previa, se recopila y, a continuación, es procesado por el mandato **execute prefetch plug-in** correspondiente, lo que puede crear un archivo que contenga descargas adicionales. Cada mandato **collect prefetch items** se trata como un punto de sincronización, lo que hace que el proceso de captación previa de la acción tenga que esperar a que los archivos se descarguen antes de continuar. Cuando los archivos estén disponibles, la acción se volverá a procesar desde el principio. Esto permite a la acción compensar por cualquier archivo que haya podido cambiar debido a condiciones alteradas de la máquina. El siguiente mandato en la acción se procesará sólo después de que el mandato **collect prefetch items** se ejecute y todos los archivos de la lista de captación previa se hayan descargado.

El mandato **end prefetch block** realiza una recopilación automática, garantizando que los mandatos de acción subsiguientes tengan los archivos necesarios a mano.

Cuando los archivos se hayan recopilado, pueden examinarse con algunos inspectores nuevos, que puede tener distintas interpretaciones, en función de si la acción está activa durante el proceso de captación previa o no. Para obtener más información sobre estos inspectores, consulte **Introducción a las descargas dinámicas**.

Sintaxis

```
collect prefetch items
```

Ejemplo

```
begin prefetch block
  parameter "ini"="{file "server_bf.ini" of site (value of setting
    "MyCustomSite") of client}"
  add prefetch item name=myPlugIn.exe sha1=12 size=12 sha256=347 size=456
    url=http://mysite/plugin.exe
  // recopilar el plug-in antes de continuar:
  collect prefetch items
  execute prefetch plug-in "{download path "myPlugIn.exe"}" /downloads
    "{parameter "ini"}" "{download path "urllist"}"
  add prefetch item {concatenation " ; " of lines of download file "urllist"}
end prefetch block
```

En este ejemplo, la sentencia **collect** garantiza que el plugin se descargue correctamente antes de continuar con el resto del bloque de captación previa.

Versión 7.2 y posteriores

copy

Copia el archivo de origen al archivo de destino denominado. Un script de acción con el mandato **copy** finaliza si el destino ya existe o si la copia falla por cualquier otro motivo (por ejemplo, cuando el archivo de destino está ocupado).

Sintaxis

```
copy <NombreArchivo_origen> <NombreArchivo_destino>
```

Donde **NombreArchivo_origen** y **NombreArchivo_destino** son los nombres de los archivos para copiar desde y a respectivamente (normalmente entre comillas).

Ejemplos

```
copy "{name of drive of windows folder}\win.com" "{name of drive of windows folder}\bigsoftware\win.com"
```

Este mandato copia el archivo win.com a la carpeta bigsoftware.

```
delete "c:\windows\system\windir.dll"
copy " __Download\windir.dll" "c:\windows\system\windir.dll"
```

Este par de mandatos de shell de acción elimina el archivo de destino (si existe) antes de llevar a cabo la acción de copia.

Versión 5.1 y posteriores

createfile until

Este mandato crea un archivo de texto denominado **__createfile** en el directorio de sitios. Le permite llenar un archivo con una serie de sentencias hasta una cadena de finalización. El mandato tiene el siguiente formato:

```
createfile until <end-delim-string>
    line 1
    line 2
    ...
end-delim-string
```

Nota: asegúrese de que las líneas etiquetadas 'line 1, line 2, .' no contengan end-delim-string inadvertidamente. En caso afirmativo, el analizador de la acción empezará a buscar mandatos de acción después de la primera instancia end-delim-string.

Sintaxis

createfile until <delimitador>

statements...

delimiter

Ejemplos

```
parameter "config" = "{pathname of folder (value of variable "tmp" of environment)}\config.txt"
createfile until end
    Operating system = {name of operating system}
    Processor count = {number of processors}
end
delete "{parameter "config"}"
copy __createfile "{parameter "config"}"
```

Define un parámetro denominado "config" que contiene el nombre de la ruta del archivo config en la carpeta tmp, crea un nuevo archivo name=value que contiene el sistema operativo y el recuento de procesador, suprime el archivo de configuración de la carpeta tmp y lo sustituye por el nuevo archivo.

Versión 6.0 y posteriores -- solo Windows

suprimir

Suprime el archivo designado. Cualquier script de acción con el mandato delete terminará si el archivo existe pero no se puede suprimir. Esto puede suceder debido a protección contra grabación o un intento de suprimir desde un CD-ROM, por ejemplo. Si el archivo no existe, sin embargo, el script de acción continuará ejecutándose.

Sintaxis

delete <nombreArchivo>

Donde **nombreArchivo** es el nombre del archivo que se suprimirá (normalmente entre comillas). La sustitución de relevancia se realiza en los argumentos de las líneas de mandato de delete action.

Ejemplos

```
delete "c:\program files\bigsoftware\module.dll"  
delete "{name of drive of windows folder}\win.com"
```

Estos ejemplos suprimen los archivos especificados. Tenga en cuenta que puede utilizar la sustitución de variables (entre llaves) para especificar los nombres de rutas.

Nota

Se recomienda escribir los nombres de archivos entre comillas para conservar los espacios de los nombres de archivo. Sin comillas, el sistema de archivos no podrá acceder a estos archivos con espacios en el nombre de ruta o de archivo.

Versión 5.1 y posteriores

download

En desuso: utilice **download as** o **download now as**.

Descarga el archivo indicado por el URL. Este mandato se incluye por motivos de retrocompatibilidad con la versión 2.0 de Client Edition, y se sigue dando soporte para manejar correctamente acciones de IBM Endpoint Manager antiguas. Para todas las demás aplicaciones, este mandato se ha reemplazado por los mandatos **download as** y **download now as**.

Después de descargarlo, el archivo se guarda en una carpeta denominada "__Download" (el nombre de la carpeta empieza por dos signos de subrayado) relativo a la carpeta local del sitio de Fixlet que ha emitido el mandato de descarga.

Si la descarga falla, el script de acción finaliza. El nombre del archivo se deriva de la parte del URL después de la última barra inclinada.

Por ejemplo, considere el mandato:

```
download ftp://ftp.microsoft.com/deskapps/readme.txt
```

El ejemplo de acción anterior descarga el archivo readme.txt desde el sitio web de Microsoft y lo guarda automáticamente en la carpeta __Download local como readme.txt.

El nombre de archivo se deriva del URL. Todo después del carácter / o \ final se considera nombre de archivo. Ocasionalmente, esto puede generar un nombre de archivo problemático, por ejemplo:

URL: `http://skdkdk.ddddd.com/cgi-bin/xyz?jjj=yyy`

genera un archivo denominado `xyz?jjj=yyy`, que no es un nombre de archivo válido. Normalmente puede encontrar una solución alternativa añadiendo un argumento ficticio al final del URL:

`http://skdkdk.ddddd.com/cgi-bin/xyz?jjj=yyy?file=/ddd.txt`

que generará un archivo denominado `ddd.txt`, que se guarda en el directorio __Download. Los mandatos **download as** y **prefetch** también se pueden utilizar para tratar esta situación.

Sintaxis

`download [option] <URL_archivo>`

Donde el prefacio **[options]** puede ser una de dos palabras clave opcionales:

open: llama a la API ShellExecute, pasando el nombre de archivo resultante una vez que la descarga se completa.

now: indica al cliente de IBM Endpoint Manager que inicie la descarga en ese punto de la ejecución de la acción, en lugar de realizar la captación previa antes de que empiece la acción. El agente intentará recopilar la descarga directamente desde el URL especificado en lugar de pasar por el sistema de retransmisores.

URL_archivo es la ubicación del archivo para descargar.

Ejemplos

`download http://download.mycompany.com/update/bfxxxx.exe`

Realiza la captación previa del archivo `bfxxxx.exe` desde el sitio de `mycompany`, y dirige el archivo descargado a la carpeta "__Download" del sitio predeterminado.

`download open http://download.bigfix.com/update/bfxxxx.exe`

Realiza la captación previa y guarda el archivo `bfxxxx.exe` en la carpeta "__Download" del sitio predeterminado y ejecuta el programa cuando la descarga se completa.

`download now http://download.mycompany.com/update/bfxxxx.exe`

Descarga el archivo `bfxxxx.exe` del sitio de `mycompany` tan pronto como el mandato se ejecuta.

`download "http://download.microsoft.com/download/prog.exe"`
`run "__Download\prog.exe"`

Este conjunto de acciones automatiza el proceso de descarga, reduciendo la aplicación de un parche ejecutable a una sola pulsación. Tenga en cuenta que el programa descargado se ejecuta desde el directorio '__Download' del sitio de

Fixlet, donde lo coloca el mandato de descarga. El directorio del sitio de Fixlet es el directorio de trabajo para todos los mandatos y el directorio `__Download` se ubica aquí.

Nota: La sustitución de relevancia **NO** se realiza en las líneas de mandato de acción de **descarga**. Esto se debe a que estas acciones se analizan mediante otros componentes que proporcionan las descargas y estos otros componentes se ejecutan en máquinas distintas que no comparten el contexto de evaluación de estos clientes. Esta restricción, sin embargo, permite a IBM Endpoint Manager realizar la captación previa de las descargas a través de una jerarquía de retransmisores para los clientes.

Versión 5.1 y posteriores

download as

Descarga el archivo indicado por el URL y permite siempre cambiarle el nombre. Después de descargarlo, el archivo se guarda en una carpeta denominada "`__Download`" (el nombre de la carpeta empieza por dos signos de subrayado) relativo a la carpeta local del sitio de Fixlet que ha emitido el mandato **download as**.

Por ejemplo, considere el mandato:

```
download as intro.txt ftp://ftp.microsoft.com/deskapps/readme.txt
```

El ejemplo de acción anterior descarga el archivo `readme.txt` desde el sitio web de Microsoft y lo guarda automáticamente en la carpeta `__Download` local como `intro.txt`. Si la descarga falla, el script de acción finaliza.

Este mandato, cuando va acompañado de **continue if** con un valor sha1 o sha256, permite realizar la captación previa del archivo.

Sintaxis

```
download as <nombre> <url>
```

Donde **nombre** es un nombre de archivo simple, sin caracteres especiales ni delimitadores de ruta. Si el nombre viola alguna de las reglas siguientes, el mandato `prefetch` no se ejecutará correctamente:

El nombre debe tener 32 caracteres o menos.

El nombre sólo debe estar compuesto por caracteres ASCII a-z, A-Z, 0-9, -, _, y puntos no iniciales.

Aquí, **url** es el URL completo del archivo especificado.

Ejemplos

```
download as myprog.exe http://www.website.com/update/prog555.exe
```

Descarga el archivo `prog555.exe` desde la carpeta especificada en el sitio web, dirige el archivo descargado a la carpeta "`__Download`" del sitio de acción y la cambia el nombre por `myprog.exe`.

```
download as patch1 http://www.download.windowsupdate.com/msdownload/update/v3-
19990518/cabpool/q307869_f323efa52f460eale5f4201b011c071ea5b95110.exe
continue if {(size of it = 813160 and sha1 of it =
"92c643875dda80022b3ce3f1ad580f62704b754f") of file "patch1" of
folder "__Download"}
```

Descarga el archivo especificado, le cambia el nombre por patch1 y continúa únicamente si el tamaño y el valor sha1 son correctos.

Nota:

la sustitución de relevancia **NO** se realiza en las líneas de mandato de acción de **download as**. Esto se debe a que estas acciones se analizan mediante otros componentes que proporcionan las descargas y estos otros componentes se ejecutan en máquinas distintas que no comparten el contexto de evaluación de estos clientes. Esta restricción, sin embargo, permite a IBM Endpoint Manager realizar la captación previa de las descargas a través de una jerarquía de retransmisores para los clientes.

Versión 6.0 y posteriores -- solo Windows

download now as

Descarga el archivo indicado por el URL y permite siempre cambiarle el nombre. Después de descargarlo, el archivo se guarda en una carpeta denominada `__Download` (el nombre de la carpeta empieza por dos signos de subrayado) relativo a la carpeta local del sitio de Fixlet que ha emitido el mandato **download now as**.

Si la descarga falla, el script de acción finaliza.

Por ejemplo, considere el mandato:

```
download now as intro.txt ftp://ftp.microsoft.com/deskapps/readme.txt
```

El ejemplo de acción anterior descarga de forma inmediata el archivo `readme.txt` desde el sitio web de Microsoft y lo guarda automáticamente en la carpeta `__Download` local como `intro.txt`.

Sintaxis

```
download now as <nombre> <url>
```

Donde **nombre** es un nombre de archivo simple, sin caracteres especiales ni delimitadores de ruta. Si el nombre viola alguna de las reglas siguientes, el mandato `prefetch` no se ejecutará correctamente:

Nombre debe tener 32 caracteres o menos.

El nombre sólo debe estar compuesto por caracteres ASCII a-z, A-Z, 0-9, -, _ y puntos no iniciales.

Aquí, **url** es el URL completo del archivo especificado.

Ejemplos

```
download now as myprog.exe http://www.website.com/update/prog555.exe
```

Descarga el archivo prog555.exe de forma inmediata desde la carpeta especificada en el sitio web, dirige el archivo descargado a la carpeta "__Download" del sitio de acción y lo denomina myprog.exe.

```
download now as patch2 http://www.download.windowsupdate.com/msdownload/update/v3-19990518/cabpool/q310507_2f3c5854999b7c58272a661d30743abca15caf5c.exe
continue if {(size of it = 845416 and sha1 of it =
"c964d4fd345b6e5fd73c2235ec75079b34e9b3d2")} of file "patch2.exe" of folder
 "__Download"}
```

Descarga de forma inmediata el archivo especificado desde el sitio web, dirige el archivo descargado a la carpeta __Download del sitio de acción y lo denomina nombre patch2. La acción continúa sólo si el tamaño y el valor sha1 son correctos.

Nota: La sustitución de relevancia **NO** se realiza en las líneas de mandato de acción de **download now as**. Esto se debe a que estas acciones se analizan mediante otros componentes que proporcionan las descargas y estos otros componentes se ejecutan en máquinas distintas que no comparten el contexto de evaluación de estos clientes. Esta restricción, sin embargo, permite a IBM Endpoint Manager realizar la captación previa de las descargas a través de una jerarquía de retransmisores para los clientes.

Versión 6.0 y posteriores -- solo Windows

end prefetch block

Marca el final de un bloque de captación previa (consulte **begin prefetch block**). Este mandato debe estar presente siempre que se especifique el mandato **begin prefetch block**. Este mandato ejecuta automáticamente un mandato **collect prefetch items**, lo que significa que todos los archivos añadidos a la lista de captación previa estarán disponibles cuando el bloque finalice.

Sintaxis

```
end prefetch block
```

Solo se puede utilizar un bloque de mandatos de captación previa en un script de acción y debe especificarse entre corchetes mediante un mandato **begin prefetch block** y un mandato **end prefetch block**.

Solo comentarios o líneas en blanco pueden preceder a este bloque de captación previa. Al procesar acciones con bloques de captación previa, **download as** y **prefetch** no están permitidos en ninguna parte del script de acción. El mandato **download now** se permite después del bloque de captación previa, pero *no antes o dentro del bloque de captación previa*.

```
begin prefetch block
  add prefetch item sha1=123 sha256=689 size=456 url=http://ms.com/downloads/hotfix123.exe
end prefetch block
wait {download path "hotfix123.exe"}
```

Este código muestra una descarga estática en un bloque de captación previa. Aunque no aproveche la sustitución de relevancia dinámica, este es el formato preferido para las descargas en las versiones 7.2 y posteriores. Tenga en cuenta que el mandato **end prefetch block** también recopila el archivo (hotfix123.exe), de modo que el mandato **wait** subsiguiente, que se ejecuta y espera para completarse, tiene la garantía de que dispondrá del archivo.

Nota: Las consolas y los clientes anteriores rechazarán los scripts de acción que utilizan la nueva funcionalidad de captación previa y los identificarán como que contienen errores de sintaxis. Los retransmisores anteriores no procesarán acciones de descarga dinámica, incluso si el servidor y los clientes pueden manejarlas.

Versión 7.2 y posteriores

execute prefetch plug-in

Este mandato pasa argumentos a un mandato designado y, a continuación, lo ejecuta. No está pensado para un ejecutable largo y el cliente sólo esperará 60 segundos para su finalización. Este mandato se puede utilizar para autenticar o ejecutar las descargas. También se puede utilizar para ejecutar lógica personalizada, que puede crear valores inspeccionables para los mandatos **add prefetch item** subsiguientes.

Para casos de uso, como la actualización de definiciones antivirus, este mandato puede utilizarse para ejecutar código que procesa un archivo para generar otro archivo que contiene un conjunto de URL que se descargarán.

Sintaxis

```
execute prefetch plug-in "nombre de ruta ejecutable" <args>
```

Donde:

"nombre de ruta ejecutable"

Es el nombre de ruta completo para ejecutar el plugin. Este mandato está diseñado para ejecutables que se ejecuten más rápido y se devuelvan inmediatamente. El cliente de IBM Endpoint Manager se bloqueará 60 segundos mientras espera que se complete el mandato; este tiempo de espera solo se puede interrumpir con una solicitud de conclusión. Después de 60 segundos, el cliente registrará un mensaje e inhabilitará el mandato. Cuando esté inhabilitado, las acciones que utilicen este mandato no se ejecutarán hasta después de que el cliente se haya reiniciado. En general, se espera que el mandato se complete más rápido-- si tarda más de dos segundos en ejecutarse, el cliente registrará un mensaje adecuado. La sustitución de relevancia se puede utilizar para especificar el nombre de la ruta.

<args>

Son los argumentos que se pasan al ejecutable.

El código de salida de la aplicación de execute prefetch plug-in es importante porque informa al cliente sobre el éxito o el fracaso de la operación, donde 0 (cero) indica que se ha realizado correctamente y todos los demás códigos de salida se tratan como anomalías y dan como resultado un intento de acción fallido. Con fines de depuración, el código de salida se registra en el registro de cliente.

Ejemplo

```
begin prefetch block
  parameter "ini_file"={file "server_bf.ini" of site (value of setting
    "MyCustomSite") of client}
  add prefetch item name=plugin.exe sha1=123 sha256=789 size=12
  url=http://mysite/myplugin.exe
  collect prefetch items
  // ejecutar el plugin para generar un manifiesto desde el archivo ini_file:
```

```
execute prefetch plug-in "{download path "plugin.exe"}" /downloads
  "{parameter "ini_file"}" "{download path "manifest"}"
add prefetch item {concatenation " ; " of lines of download file "manifest"}
end prefetch block
```

Este ejemplo descarga un plugin que procesa el archivo ini_file para generar un manifiesto.

Versión 7.2 y posteriores

extract

Extrae los archivos especificados en la carpeta de descarga (__Download) y deja los resultados en la misma carpeta.

Un archivo de archivado es similar a un archivo tar comprimido. IBM Endpoint Manager utiliza una herramienta denominada Archivewriter para crear el archivo. Esto puede ser útil para copiar un directorio completo en un sistema, lo cual suele ser necesario para programas de instalación que contienen varios archivos junto con un ejecutable de configuración. Hay un asistente en la consola de IBM Endpoint Manager que facilita la distribución de directorios que utilizan esta clase de archivo.

Como en la versión 8.2, este mandato también permite especificar un directorio objetivo con un segundo argumento opcional.

Sintaxis

```
extract <archivo de archivado> [<directorio_destino>]
```

Donde el directorio de destino es opcional y toma como valor predeterminado el directorio __download.

Ejemplos

```
extract InstallMyApp.zip
```

Extrae los archivos componentes de InstallMyApp en la carpeta __Download, coloca los resultados de nuevo en la carpeta __Download y suprime el archivo InstallMyApp original.

```
extract InstallMyApp.zip "d:/temp"
```

Igual que el anterior, pero especificando un directorio de destino.

Nota: No puede haber comillas alrededor del nombre de archivo, incluso aunque haya espacios en blanco en el nombre. Esto es para ser coherente con sistemas que permiten comillas como caracteres de nombre de archivo válidos.

Versión 5.1 y posteriores

folder create

Crea la carpeta designada. Cualquier script de acción con el mandato create terminará si no se puede crear la carpeta. Esto puede suceder debido a protección contra grabación o un intento de grabar en un CD-ROM, por ejemplo. También terminará si la ruta ya existe, pero no hace referencia a una carpeta.

Sintaxis

folder create <nombreCarpeta>

Donde **nombreCarpeta** es el nombre de la carpeta que se creará (normalmente entre comillas). La sustitución de relevancia se realiza en los argumentos de las líneas de mandato de folder create.

Ejemplos

```
folder create "c:\program files\bigsoftdir"  
folder create "{name of drive of windows folder}\Extras"
```

Estos ejemplos crean las carpetas especificadas. Tenga en cuenta que puede utilizar la sustitución de variables (entre llaves) para especificar los nombres de rutas.

Nota

Se recomienda escribir nombres de carpeta entre comillas para conservar los espacios.

Versión 8.0

folder delete

Suprime la carpeta designada. Este mandato es recursivo, suprime todos los archivos y carpetas contenidos. Un script de acción con el mandato delete terminará si la carpeta existe pero no se puede suprimir. Esto puede suceder debido a protección contra grabación o un intento de suprimir desde un CD-ROM, por ejemplo. Esta acción también fallará (y dejará el contenido de la carpeta en una condición indeterminada) cuando el contenido de la carpeta esté ocupado, como puede suceder cuando un archivo dentro de la carpeta está siendo utilizado por otra aplicación.

Si la carpeta no existe, sin embargo, el script de acción continuará ejecutándose.

Sintaxis

folder delete <nombreCarpeta>

Donde **nombreCarpeta** es el nombre de la carpeta que se suprimirá (normalmente entre comillas para conservar espacios). La sustitución de relevancia se realiza en los argumentos de las líneas de mandato de delete folder.

Ejemplos

```
folder delete "c:\program files\bigsoftdir"  
folder delete "{name of drive of windows folder}\Temp"
```

Estos ejemplos suprimen las carpetas especificadas. Tenga en cuenta que puede utilizar la sustitución de variables (entre llaves) para especificar los nombres de rutas.

Atención: Este mandato es recursivo y puede suprimir todos los archivos en el cliente.

Versión 8.0

move

Desplaza el archivo de origen al archivo de destino designado. Este mandato también ofrece al autor de la acción la posibilidad de cambiar el nombre de un archivo. Un script de acción con el mandato `move` finaliza si el destino ya existe, si el archivo de origen no existe o si la operación de traspaso falla por cualquier otro motivo.

Sintaxis

```
move <NombreArchivo_origen> <NombreArchivo_destino>
```

Donde **NombreArchivo_origen** y **NombreArchivo_destino** son los nombres de los archivos para mover desde y a respectivamente (normalmente entre comillas).

Ejemplos

```
move "c:\program files\bigsoftware\module.dll" "c:\temp\mod.dll"
```

Este mandato mueve y renombra el archivo `mod.dll`. Tenga en cuenta que las comillas son necesarias para nombres de archivos y carpetas que incluyen espacios.

```
delete "c:\updates\q312456.exe"  
move "__Download\q312456.exe" "c:\updates\q312456.exe"
```

La línea de mandatos anterior primero suprime el archivo, a continuación lo vuelve a colocar en su lugar desde otra ubicación.

Versión 5.1 y posteriores

prefetch

El mandato `prefetch` permite descargar un archivo antes de que se inicie la acción. No necesita una sentencia **continue if** que coincida para descargar el archivo y comprobarlo por adelantado. Se prefiere el mandato de captación previa sobre el mandato **download**.

Por ejemplo, considere el mandato:

```
prefetch a.exe sha1:0123456789012345678901234567890123456789  
sha256:0a1b2345678901234567g8901j234e5678g90y12r3456789345678923167e3se  
size:11723 http://x/z.exe
```

El ejemplo de acción anterior realiza la captación previa del archivo `z.exe` desde el sitio especificado y lo guarda automáticamente en la carpeta `__Download` local como `a.exe`.

Sintaxis

```
prefetch <nombre> sha1:<valor> size:<valor> sha256:<valor> <url>
```

Donde **nombre** es un nombre de archivo simple, sin caracteres especiales ni delimitadores de ruta. Si el nombre viola alguna de las reglas siguientes, el mandato de captación previa no se ejecutará correctamente:

Nombre debe tener 32 caracteres o menos.

El nombre sólo debe estar compuesto por caracteres ASCII a-z, A-Z, 0-9, -, _, y puntos no iniciales.

Aquí, **sha1:valor** o **sha256:valor** representan el valor de algoritmo de hash seguro; **size:valor** representa el tamaño del archivo en bytes y **url** representa la ubicación del sitio, incluido el nombre de archivo.

Ejemplo

```
prefetch patch3 sha1:92c643875dda80022b3ce3f1ad580f62704b754f
size:813160 http://www.download.windowsupdate.com/msdownload
/update/v3-19990518/cabpool/
q307869_f323efa52f460ea1e5f4201b011c071ea5b95110.exe
```

Esta línea de código realiza la captación previa del archivo determinado de la carpeta especificada en el sitio web, dirige el archivo descargado a la carpeta "__Download" del sitio de acción y le cambia el nombre por patch3.

```
if {name of operating system = "WinXP"}
  prefetch patch.exe sha1:92c643875dda80022b3ce3f1ad580f62704b754f
  size:813160 http://www.download.windowsupdate.com/msdownload
  /update/v3-19990518/cabpool/
  q307869_f323efa52f460ea1e5f4201b011c071ea5b95110.exe
else
  prefetch patch.exe sha1:c964d4fd345b6e5fd73c2235ec75079b34e9b3d2
  size:845416 http://www.download.windowsupdate.com/msdownload
  /update/ v3-19990518/cabpool/
  q310507_2f3c5854999b7c58272a661d30743abca15caf5c.exe
endif
utility __Download\patch.exe
wait __Download\patch.exe
```

Este código realiza la captación previa de un archivo en función del sistema operativo, guarda el archivo en la caché de utilidades como patch.exe y espera a que finalice para continuar con la acción.

Nota: La sustitución de relevancia **NO** se lleva a cabo en las líneas de mandato de acción **prefetch**. Esto se debe a que estas acciones se analizan mediante otros componentes que proporcionan las descargas y estos otros componentes se ejecutan en máquinas distintas que no comparten el contexto de evaluación de estos clientes. Esta restricción, sin embargo, permite a IBM Endpoint Manager realizar la captación previa de las descargas a través de una jerarquía de retransmisores para los clientes.

Versión 6.0 y posteriores -- solo Windows

relay select

El mandato relay select obliga al cliente de IBM Endpoint Manager a seleccionar el retransmisor más cercano, si hay alguno disponible. Este mandato emite una solicitud al cliente para que lleve a cabo una selección de retransmisor en la siguiente ocasión y siempre lo logra de inmediato, independientemente del éxito o fracaso de la selección del retransmisor pendiente.

Sintaxis

relay select

Ejemplos

relay select

Este mandato indica al cliente de IBM Endpoint Manager que busque y se conecte al retransmisor más cercano.

Versión 5.1 y posteriores

utility

El mandato `utility` puede utilizarse para colocar los programas que se utilizan con más frecuencia en una memoria caché especial. Por ejemplo:

```
utility __Download/RunQuiet.exe
```

Este mandato coloca el programa común **RunQuiet** en la caché de utilidades para evitar que se descargue varias veces.

Los clientes 6.0 mantienen dos memorias caché de disco; una para los programas de utilidad y otra para las cargas útiles de acción. Los archivos que llegan a la caché de cargas útiles de acción no sacarán archivos de la caché de utilidades ni viceversa.

Los clientes 6.0 utilizan el valor sha1 o sha256 de una descarga de acción para localizar cualquier programa de utilidad coincidente (por ejemplo 'RunQuiet') que ya exista en el cliente.

Se crea una carpeta específica para contener las descargas a medida que se realiza su captación previa. Los archivos existentes que coinciden con los valores sha1 o sha256 no tienen que volver a descargarse. La captación previa de cualquier otro archivo se realizará desde el retransmisor principal. Cuando todas las descargas estén disponibles en el cliente, los archivos se moverán de la carpeta específica de la acción (se trata de un cambio desde el comportamiento del cliente 6.0 previo) a la carpeta `__Download` del sitio de acción y la acción se iniciará.

Cuando la acción se completa, los archivos dejados en la carpeta `__Download` que se hayan captado previamente con sha1 o sha256 serán candidatos para el almacenamiento en caché de programa de utilidad. En estos archivos se volverán a calcular los valores sha1 o sha256 y los archivos que tengan valores sha1 o sha256 coincidentes se podrán mover a la memoria caché de programa de utilidad.

Un esquema utilizado menos recientemente se utiliza para mantener la caché dentro de los límites de tamaño. El límite de la caché se puede superar debido a archivos únicos y solo en intervalos cortos.

Sintaxis

```
utility <nombre_ruta>
```

Ejemplo

```
prefetch patch.exe sha1:92c643875dda80022b3ce3f1ad580f62704b754f
__size:813160 http://www.download.windowsupdate.com/msdownload/update/v3-
__19990518/cabpool/q307869_f323efa52f460eale5f4201b011c071ea5b95110.exe
utility __Download\patch.exe
wait __Download\patch.exe
```

Este ejemplo realiza la captación previa de un archivo, guarda el archivo en la caché de utilidades como patch.exe y espera a que finalice para continuar con la acción.

Versión 6.0 y posteriores -- solo Windows

Capítulo 5. Establecimiento de mandatos

establecer

Los parámetros de configuración se denominan valores que se pueden aplicar a sitios de Fixlet individuales o a los equipos cliente. Cada valor tiene un tiempo asociado a él. Esta indicación de fecha y hora se utiliza para establecer la prioridad-- el último valor tendrá prioridad sobre cualquier valor anterior.

Los operadores de la consola de IBM Endpoint Manager pueden crear y propagar valores. Los valores emitidos por la consola se etiquetarán con la fecha y hora actual. Los valores se separan en grupos, incluyendo uno para cada sitio y uno para el cliente. Cada grupo de valores es independiente de los demás y es persistente en el cliente.

Las acciones en Fixlets también pueden crear valores y normalmente utilizan la sustitución {now}, que se evalúa cuando se ejecuta la acción. Puede examinar estos valores y sus indicaciones de fecha y hora con inspectores como "effective date of <setting>" (consulte la Guía de inspectores para obtener más información).

Sintaxis

setting "<nombre>"="value" on "<fecha>" for client

setting "<nombre>"="value" on "<fecha>" for site "<nombresitio>"

Donde **nombre=value** describe el valor y **fecha** es una indicación de fecha y hora utilizada para establecer la prioridad. Estos puede establecerse para equipos **cliente** o para un **sitio** designado.

Ejemplos

```
setting "name"="Bob" on "31 Jan 2007 21:09:36 gmt" for client
```

Establece la variable de nombre en Bob en la máquina cliente con una indicación de fecha y hora MIME proporcionada por la consola cuando se creó este valor. Reemplazará cualquier otro valor de nombre con fecha anterior.

```
setting "preference"="red" on "{now}" for site "color_site"
```

Al ejecutar la acción que contiene este mandato, {now} se evalúa como una fecha/hora MIME y se sustituye en la cadena. Este mandato establece la variable "preference" en "red" para el sitio de Fixlet especificado. Tenga en cuenta que a menos que haya varios sitios con el mismo nombre, puede especificar el sitio sin el URL de recopilación completo. Puede tener un valor "preference" distinto en cada sitio.

```
setting "time"="{now}" on "{now}" for current site
```

Establece de forma inmediata la variable de hora a la hora actual en el sitio actual.

```
setting "division"="%22design group%22" on "15 Mar 2007 17:05:46 gmt" for client
```

Este ejemplo utiliza %xx para indicar caracteres especiales por su equivalente hexadecimal. En este caso, %22 comprende el valor de la variable entre comillas dobles.

Nota

Cuando un cliente se restablece, las fechas efectivas de los valores se eliminan y cualquier mandato de valor subsiguiente las sustituirá. Hay varias formas de restablecer los clientes, incluidas las colisiones de ID de equipo (con mayor frecuencia causado al incluir accidentalmente el ID de equipo en una imagen que se copia en varios sistemas), el cambio de un encabezado de sitio de acciones a un nuevo servidor o indicando al cliente que recopile un nuevo ID.

Las acciones que se ejecutan a continuación establecerán una nueva fecha efectiva, pero el valor *values* que será el mismo que antes de la acción de restablecer. Los valores se conservan porque contienen información como, por ejemplo, las selecciones de retransmisores. De este modo, cuando se produce un restablecimiento de despliegue, no tiene que emitir nuevas acciones para restablecer su estructura de retransmisores de red.

Versión 5.1 y posteriores

setting delete

Esta acción suprime una variable de valor designada en el equipo cliente. Incluye una indicación de fecha y hora que se comparará con la indicación de fecha y hora de la configuración original. Si la fecha de supresión es posterior a la fecha del valor, el valor se suprimirá. De lo contrario, el mandato delete se ignorará.

Sintaxis

setting delete "<nombre>" on "<fecha>" for client

setting "<nombre>" on "<fecha>" for site "<url_sitio>"

Donde **nombre** describe el valor que se suprimirá, y **fecha** es cuando se suprimirá el valor. Los valores pueden suprimirse en los equipos **cliente** o en un **sitio** designado.

Ejemplos

```
setting delete "name" on "14 Apr 2007 21:09:36 gmt" for client
```

Suprime la variable "name" en la máquina cliente si la indicación de fecha y hora es posterior al valor de tiempo correspondiente. De lo contrario, el mandato delete se ignorará.

```
setting delete "abc" on "{now}" for site "siteurl"
```

Suprime de forma inmediata la variable "abc" en el sitio especificado.

```
setting delete "abc" on "{now}" for current site
```

Suprime de forma inmediata la variable "abc" en el sitio actual.

Versión 5.1 y posteriores

Capítulo 6. Mandatos de registro

regdelete

Suprime un valor de clave de registro del nombre proporcionado, independientemente de si existe actualmente o no.

Sintaxis

```
regdelete "<clave de registro>" "<nombre de valor>"
```

Donde **clave de registro** es el nombre de la clave y **nombre de valor** es el valor en la clave de registro que desea eliminar.

Ejemplo

```
regdelete "[HKEY_CLASSES_ROOT\ShellScrap]" "NeverShowExt"
```

Suprime el valor NeverShowExt de la clave de registro especificada.

Notas

Este mandato es solo de Windows. Hará que un script de acción termine en un agente Unix.

Para suprimir una clave de registro que no esté vacía y todos sus subclaves, deberá crear un archivo, por ejemplo del.reg, que tiene este aspecto:

```
REGEDIT4  
[-HKEY_CURRENT_USER\keep\removethisandbelow]
```

Debe haber tres líneas en este archivo: la última línea debe estar en blanco. Fíjese en el guión largo (-) delante de la ruta de registro.

Ahora puede ejecutar una acción como esta:

```
regedit /s del.reg
```

Cuando se ejecuta esta acción, la clave denominada removethisandbelow, junto con todas sus subclaves, se suprime. Puede utilizar el mandato **appendfile** para crear este archivo .reg.

Si la clave especificada ya no existe, este mandato la creará.

Versión 5.1 y posteriores -- solo Windows

regset

Establece una clave de registro para el nombre y el valor proporcionados. Si la clave ya no existe, este mandato la crea con el valor de inicio.

Sintaxis

```
regset "<clave de registro>" "<nombre de valor>"=<valor>
```

Donde **clave de registro** es la clave de interés y **nombre de valor** es el valor de clave que se definirá en **valor**. Estos valores se especifican tal como se encuentran en un archivo de registro REGEDIT4, de acuerdo con las reglas para Regedit, el programa de Windows que edita el registro. Los valores de cadena están delimitados por comillas, y el entero de 4 bytes estándar (dword) se identifica utilizando dword: seguido por el valor numérico especificado en hexadecimal (con los ceros iniciales) como se muestra a continuación.

Ejemplos

```
regset "[HKEY_CURRENT_USER\Software\Microsoft\Office\9.1\Word\Security]"
  "Level"=dword:00000002
```

Este ejemplo establece el valor de Level del registro especificado para un 2 de palabra doble.

```
regset "[HKEY_CURRENT_USER\Software\BigCorp Inc.]" "testString"="bob"
```

Este ejemplo establece el valor de testString de la clave de registro especificada en bob.

```
regset "[HKEY_CLASSES_ROOT\ShellScrap]" "AlwaysShowExt"=""
```

Este ejemplo borra los datos del valor de registro especificado.

Notas

Este mandato es solo de Windows. Hará que un script de acción termine en un agente Unix.

Fíjese en estos ejemplos que se utilizan corchetes [] para especificar el nombre de la clave de registro. De nuevo, de acuerdo con las reglas para los archivos de registro REGEDIT4. Esta sintaxis es necesaria para el mandato RegSet, pero no para inspectores de registro.

Cuando utilice el mandato regset, tenga en cuenta que el cliente de IBM Endpoint Manager crea el archivo .reg dinámicamente, que tendría que haber creado manualmente para actualizar el registro y, a continuación, ejecuta el archivo. Una de las reglas del archivo .reg consiste en que cualquier \ en el campo **valor** tiene que aparecer como barras inclinadas dobles, es decir, \\. De manera que si intentaba asignar el valor SourcePath2 de la clave de registro HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion a c:\I386, el mandato que definiría tendría el siguiente aspecto:

```
regset "[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion]"
  "SourcePath2"="c:\\I386"
regset "[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion]"
  "SourcePath2"={escape of "c:\I386"}
```

El último ejemplo utiliza la cláusula de relevancia **escape** para convertir automáticamente barras invertidas en barras invertidas dobles.

En los casos en los que necesita emitir muchos mandatos regset, puede considerar la posibilidad de utilizar los mandatos **appendfile** o **createfile until** para crear un archivo regedit con el formato correcto y, a continuación, ejecutar regedit de forma silenciosa:

```
Createfile until end-reg-edit-commands
  REGEDIT4
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion]
  "SourcePath1"="c:\\I386"
```

```
"SourcePath2"="{escapes of pathname of windows folder}"  
end-reg-edit-commands  
move __createfile setup.reg  
wait regedit /s setup.reg
```

Si la clave especificada no existe, este mandato la creará.

Versión 5.1 y posteriores -- solo Windows

Capítulo 7. Mandatos Wow64

action uses wow64 redirection

Este mandato permite al cliente salir del mundo de 32 bits creado para él por el recurso **Windows On Windows64** (Wow64), incorporado en las nuevas versiones de 64 bits del sistema operativo Windows, incluidos Windows 2003 x64 y Windows XP Pro x64.

Cuando este mandato se ejecuta en una acción en un sistema operativo de 64 bits con un valor de **true**, el cliente permite la redirección de Wow64 en cualquier mandato subsiguiente que incluya nombres de archivo. Este estado continúa hasta que la acción se completa o el cliente ejecuta el mandato **action uses wow64 redirection false**.

Puede utilizar la **sustitución de relevancia** para proporcionar el valor <true|false>. La redirección del sistema de archivos que proporciona Wow64 se inhabilita mediante Wow64DisableWow64FsRedirection y se vuelve a habilitar con el API de Windows Wow64RevertWow64FsRedirection.

Entre los mandatos afectados por este valor se incluyen:

- dos
- run, wait, rundetached, waitdetached, runhidden, waithidden
- delete, copy, move, open

Sintaxis

action uses wow64 redirection <true|false>

Ejemplo

```
action uses wow64 redirection true
```

Este ejemplo se activa la redirección de Wow64.

```
action uses wow64 redirection false
```

Este ejemplo desactiva la redirección de Wow64.

Notas

Este mandato es solo de Windows. Hará que un script de acción termine en un agente Unix.

Versión 6.0 y posteriores -- solo Windows

regdelete64

Regdelete64 utiliza la misma sintaxis que el mandato **regdelete**, pero coloca una llamada a Wow64DisableWow64FsRedirection antes de iniciar la versión de 64 bits de Regedit para establecer el registro, que le permitirá utilizar el registro de 64 bits disponible en máquinas de 64 bits. Este mandato suprime un valor de clave de

registro del nombre proporcionado. Si el valor ya no existe, este mandato fallará y todos los mandatos subsiguientes no se ejecutarán.

Sintaxis

```
regdelete64 "<clave de registro>" "<nombre de valor>"
```

Donde **clave de registro** es el nombre de la clave y **nombre de valor** es el valor en la clave de registro que desea eliminar.

Ejemplo

```
regdelete64 "[HKEY_CLASSES_ROOT\ShellScrap]" "NeverShowExt"
```

Suprime el valor NeverShowExt de la clave de registro especificada.

Notas

Este mandato es solo de Windows. Hará que un script de acción termine en un agente Unix.

Si la clave especificada ya no existe, este mandato la creará.

Versión 6.0 y posteriores -- solo Windows

regset64

Regset64 utiliza la misma sintaxis que el mandato **regset**, pero coloca una llamada a Wow64DisableWow64FsRedirection antes de iniciar la versión de 64 bits de Regedit para establecer el registro. Este le permite utilizar el registro de 64 bits nativo para definir una clave de registro para el nombre y el valor proporcionados. Si la clave ya no existe, este mandato la crea con el valor de inicio.

Sintaxis

```
regset64 "<clave de registro>" "<nombre de valor>"=<valor>
```

Donde **clave de registro** es la clave de interés y **nombre de valor** es el valor de clave que se definirá en **valor**. Estos valores se especifican tal como se encuentran en un archivo de registro REGEDIT4, de acuerdo con las reglas para Regedit, el programa de Windows que edita el registro. Los valores de cadena están delimitados por comillas, y el entero de 4 bytes estándar (dword) se identifica utilizando dword: seguido por el valor numérico especificado en hexadecimal (con los ceros iniciales) como se muestra a continuación.

Ejemplos

```
regset64 "[HKEY_CURRENT_USER\Software\Microsoft\Office\9.1\Word\Security]"  
"Level"=dword:00000002
```

Este ejemplo establece el valor de Level del registro especificado para un 2 de palabra doble.

```
regset64 "[HKEY_CURRENT_USER\Software\BigCorp Inc.]" "testString"="bob"
```

Este ejemplo establece el valor de testString de la clave de registro especificada en bob.

```
regset64 "[HKEY_CLASSES_ROOT\ShellScrap]" "AlwaysShowExt"=""
```

Este ejemplo borra los datos del valor de registro especificado.

Nota:

Este mandato es solo de Windows. Hará que un script de acción termine en un agente UNIX.

Fíjese en estos ejemplos que se utilizan corchetes [] para especificar el nombre de la clave de registro. De nuevo, de acuerdo con las reglas para los archivos de registro REGEDIT4. Esta sintaxis es necesaria para el mandato RegSet, pero no para inspectores de registro.

Cuando utilice el mandato regset64, tenga en cuenta que el cliente de IBM Endpoint Manager crea el archivo .reg dinámicamente, que tendría que haber creado manualmente para actualizar el registro y, a continuación, ejecuta el archivo. Una de las reglas del archivo .reg consiste en que cualquier \ en el campo **valor** tiene que aparecer como barras inclinadas dobles, es decir \\.

De manera que si intentaba asignar el valor SourcePath2 de la clave de registro HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion to c:\I386, el mandato que definiría tendría el siguiente aspecto:

```
regset64 "[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion]"
"SourcePath2"="c:\I386"
regset64 "[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion]"
"SourcePath2"={escape of "c:\I386"}
```

El último ejemplo utiliza la cláusula de relevancia **escape** para convertir automáticamente barras invertidas en barras invertidas dobles.

Si la clave especificada ya no existe, este mandato la creará.

Versión 6.0 y posteriores -- solo Windows

script64

Script64 utiliza la misma sintaxis que el mandato **script**, pero coloca una llamada a Wow64DisableWow64FsRedirection antes de ejecutar el script. Esto le permite emitir un mandato de script de 64 bits nativo, pasando por alto el entorno de 32 bits de Windows creado sobre procesadores de 64 bits.

La palabra clave de script ejecuta un script externo (creado en un lenguaje de scripts como JavaScript o Visual Basic) con el nombre proporcionado. El script de acción que contiene la palabra clave terminará si no está instalado el motor de scripts adecuado o si el script no se puede ejecutar. La línea siguiente de la acción Shell Command no se ejecuta hasta que el script especificado finaliza.

Sintaxis

script64 <nombre de script>

Ejemplo

```
script64 attrib.vbs
```

Ejecuta el script de Visual BASIC attrib.vbs en el modo nativo de 64 bits.

Notas

Este mandato es solo de Windows. Hará que un script de acción termine en un agente Unix.

En un sistema Windows, este mandato tiene el mismo efecto que llamar a `Wow64DisableWow64FsRedirection` y, a continuación, emitir una sentencia `wscript "scriptName"` desde Windows.

Versión 6.0 y posteriores -- solo Windows

Capítulo 8. Mandatos de derechos administrativos

administrator add

Este mandato le permite designar personas específico para administrar clientes de IBM Endpoint Manager específicos. Este paso se lleva a cabo mediante un valor con una fecha efectiva, pasada como parámetro. La fecha no es opcional. Las pruebas de fecha efectiva son las mismas que para los **valores** normales.

Sintaxis

administrator add <nombre de administrador> on <fecha>

Ejemplo

```
administrator add "bob" on "21 Aug 2002 17:39:14 gmt"
```

Permite al operador de la consola denominado bob disponer de derechos de administrador en el equipo(s) de destino, efectivo en la fecha indicada.

Versión 5.1 y posteriores

administrator delete

Este mandato le permite eliminar los derechos administrativos del administrador especificado. Este paso se lleva a cabo mediante un valor con una fecha efectiva, pasada como parámetro. La fecha no es opcional. Las pruebas de fecha efectiva son las mismas que para los **valores** normales.

Sintaxis

administrator delete <nombre de administrador> on <fecha>

Ejemplo

```
administrator delete "bob" on "21 Aug 2002 17:39:14 gmt"
```

Elimina los derechos administrativos del operador de la consola denominado bob, en la fecha indicada.

Versión 5.1 y posteriores

Capítulo 9. Mandatos de mantenimiento de cliente de BigFix

module add

Añade el archivo de biblioteca de inspectores especificado para el conjunto de las bibliotecas de inspectores que utilizará el cliente. Al sustituir una biblioteca de inspectores, debe especificarla en un mandato **module delete**, así como en el mandato **module add**. Para completar la acción debe emitir un mandato **module commit**.

Sintaxis

```
module add "<nombre de módulo>"
```

Ejemplo

```
module add "dellinspect.dll"
```

Nota

Sólo para uso interno.

Versión 5.1 y posteriores

module commit

Los mandatos **add** y **delete** definen la fase de validación de cambios de las bibliotecas de inspectores. El mandato **commit** realiza la supresión y la adición.

Sintaxis

```
module commit
```

Ejemplo

```
delete "dellinspect.dll"  
copy "{pathname of client folder of site \"dell\"}\\dellinspect.dll" "dellinspect.dll"  
module add "dellinspect.dll"  
module commit
```

Nota

Sólo para uso interno.

Versión 5.1 y posteriores

module delete

Marca el archivo de biblioteca de inspectores especificado para su supresión. Para completar la acción debe emitir un mandato **module commit**.

Sintaxis

module delete "<nombre de módulo>"

Ejemplo

```
module delete "inspectors.dll"
```

Nota

Sólo para uso interno.

Versión 5.1 y posteriores

Capítulo 10. Bloqueo de mandatos

action lock indefinite

Activa el bloqueo de acciones, a partir de la fecha efectiva, que no caduca nunca. La fecha está en formato de hora MIME (como en 15 Mar 2007 12:42:51 -0700).

Sintaxis

`action lock indefinite "<fecha efectiva>"`

Ejemplo

```
action lock indefinite "{now}"
```

Activa inmediatamente el bloqueo de acciones.

Versión 5.1 y posteriores

action lock until

Bloquea acciones desde la fecha efectiva hasta la fecha de caducidad. La fecha de caducidad está en formato de hora MIME (como en 19 Jul 2007 12:42:51 -0700). Puede utilizar la sustitución con un Inspector como {now}, que evaluará la fecha y la insertará en la cadena.

Sintaxis

`action lock until "<fecha de caducidad>" "<fecha efectiva>"`

Ejemplo

```
action lock until "{now + 3*days}" "{now}"
```

Bloquea las acciones de forma inmediata, desbloqueándolas a los tres días.

```
action lock until "{apparent registration server time + 10 * minutes}" "{apparent registration server time}"
```

Bloquea las acciones durante 10 minutos, utilizando el **tiempo del servidor de registro aparente** actual, que se basa en la última vez que el cliente se registró con el servidor.

Versión 5.1 y posteriores

action unlock

Desbloquea el cliente para que actúe sobre cualquier acción. El campo de fecha efectiva se utiliza para asegurarse de que el bloqueo y desbloqueo de acciones se producen en el orden en el que se han creado. La fecha está en formato de hora MIME (como en 29 Nov 2008 12:42:51 -0700).

Sintaxis

action unlock "<fecha efectiva>"

Ejemplo

action unlock "{now}"

Desbloquea las acciones de forma inmediata.

Versión 5.1 y posteriores

Capítulo 11. Mandatos de mantenimiento del sitio

site force evaluation

Hace que el cliente vuelva a evaluar todos los Fixlets para el sitio. Esto resulta útil después de actualizar archivos o la configuración, para asegurarse de que se vuelva a calcular la relevancia de Fixlet para todo el sitio tan pronto como sea posible.

Sintaxis

site force evaluation

Ejemplo

```
site force evaluation
```

Versión 5.1 y posteriores

site gather schedule disable

Este mandato inhabilita la recopilación planificada desde el sitio actual. No es efectivo para los sitios de acción.

Sintaxis

site gather schedule disable

Ejemplo

```
site gather schedule disable
```

Versión 5.1 y posteriores

site gather schedule manual

Este mandato habilita la recopilación manual desde el sitio actual. No es efectivo para los sitios de acción.

Sintaxis

site gather schedule manual

Ejemplo

```
site gather schedule manual
```

Versión 5.1 y posteriores

site gather schedule publisher

Este mandato establece la planificación para recopilar desde el sitio actual al especificado en el encabezado del sitio.

Sintaxis

site gather schedule publisher

Ejemplo

site gather schedule publisher

Versión 5.1 y posteriores

site gather schedule seconds

Este mandato establece la planificación para recopilar desde el sitio de origen al número de segundos especificado.

Sintaxis

site gather schedule seconds <segundos>

Ejemplo

site gather schedule seconds 360

Establece la planificación de recopilación del sitio a 6 minutos.

Versión 5.1 y posteriores

subscribe

Suscribe al cliente en el sitio identificado en el archivo de encabezado. La consola de IBM Endpoint Manager proporciona el diálogo **Administrar sitios** para automatizar la adición de sitios.

Sintaxis

subscribe "<nombre de archivo de encabezado>"

Ejemplo

subscribe "__Download\Sitename.fxm"

Nota

En IBM Endpoint Manager, este mandato devuelve un error, a menos que se ejecute como una acción en el sitio de acciones maestro. El mandato es útil para suscribirse a los clientes de sitios de Enterprise Fixlet y para actualizar el archivo de encabezado de sitio de acciones.

Versión 5.1 y posteriores

darse de baja

Da de baja automáticamente del sitio de Enterprise Fixlet actual.

Sintaxis

unsubscribe

Ejemplo

darse de baja

Versión 5.1 y posteriores

Capítulo 12. Comentarios

barra inclinada doble

Las líneas que empiezan por // son comentarios y se ignoran durante la ejecución de la acción.

Sintaxis

//

Ejemplo

```
//El siguiente mandato sustituirá al archivo en la unidad C:  
copy "{name of drive of windows folder}\win.com" "{name of drive of windows  
folder}\bigsoftware\win.com"
```

Las barras inclinadas dobles le permiten comentar sus scripts de acción.

Versión 5.1 y posteriores

Apéndice. Asistencia

Para obtener más información sobre este producto, consulte los siguientes recursos:

- IBM Knowledge Center
- Sitio de soporte de IBM Endpoint Manager
- Wiki de IBM Endpoint Manager
- Base de conocimiento
- Foros y comunidades

Avisos

Esta información se ha desarrollado para productos y servicios ofrecidos en EE.UU.

Puede que IBM no ofrezca en otros países los productos, servicios o características que se describen en este documento. Consulte a su representante local IBM para obtener información sobre los productos y servicios que están disponibles en su área en la actualidad. Las referencias a programas, productos o servicios de IBM no pretenden establecer o implicar que sólo puedan utilizarse los productos, programas o servicios de IBM. En su lugar, se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no infrinja ninguno de los derechos de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patentes pendientes que cubran el tema principal descrito en este documento. La posesión de este documento no le otorga ninguna licencia sobre dichas patentes. Puede enviar sus consultas sobre licencias, por escrito, a:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
Estados Unidos de América*

Para las consultas de licencias relativas a información sobre el juego de caracteres de doble byte (DBCS), póngase en contacto con el Departamento de propiedad intelectual de IBM en su país o envíe las consultas, por escrito, a:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokio 103-8510, Japón*

El siguiente párrafo no se aplica al Reino Unido ni a ningún otro país donde estas disposiciones estén en contradicción con la legislación local:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN GARANTÍAS DE NINGÚN TIPO, NI EXPLÍCITAS NI IMPLÍCITAS, INCLUIDAS PERO SIN LIMITARSE A LAS MISMAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN, COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO. Algunos estados no permiten la renuncia a garantías explícitas o implícitas en determinadas transacciones, por lo que puede que esta declaración no sea aplicable en su caso.

Esta información podría contener imprecisiones técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información aquí contenida; estos cambios se incorporarán en nuevas ediciones de la publicación. IBM puede realizar mejoras y/o cambios en el producto productos y/o en el programa o programas descritos en esta publicación en cualquier momento y sin previo aviso.

Todas las referencias hechas en este documento a sitios web que no son de IBM se proporcionan únicamente para su información y no representan en modo alguno una recomendación de dichos sitios web. Los materiales de estos sitios web no forman parte de los materiales de IBM para este producto, y el uso que se haga de estos sitios web será responsabilidad del usuario.

IBM puede utilizar o distribuir cualquier información que se le proporcione en la forma que considere adecuada, sin incurrir por ello en ninguna obligación para con el remitente.

Los propietarios de licencias de este programa que deseen obtener información sobre el mismo con el fin de permitir: (i) el intercambio de información entre programas creados independientemente y otros programas (incluido éste) y (ii) el uso mutuo de la información que se ha intercambiado, deben ponerse en contacto con:

*IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758, Estados Unidos*

Dicha información puede estar disponible, sujeta a los términos y condiciones adecuados, incluido en algunos casos el pago de una tarifa.

IBM proporciona el programa bajo licencia que se describe en esta información y todo el material bajo licencia disponible bajo los términos del acuerdo IBM Customer Agreement, IBM International Program License Agreement o de cualquier acuerdo equivalente entre las partes.

Los datos sobre rendimiento aquí incluidos se han determinado en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar considerablemente. Algunas mediciones pueden haberse realizado en sistemas a nivel de desarrollo y no existe ninguna garantía de que estas mediciones serán las mismas en los sistemas de disponibilidad general. Además, es posible que algunas mediciones se haya estimado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables para su entorno específico.

La información referente a productos no IBM se ha obtenido de los proveedores de dichos productos, sus anuncios publicados o de otras fuentes disponibles públicamente. IBM no ha probado esos productos y no puede confirmar la precisión del rendimiento, compatibilidad ni ninguna otra afirmación referente a productos que no sean de IBM. Las preguntas sobre las posibilidades de los productos no IBM deben dirigirse a los proveedores de dichos productos.

Todas las declaraciones relativas a la dirección o intenciones futuras de IBM pueden cambiar o ser retiradas sin aviso, y representan sólo propósitos y objetivos.

Todos los precios de IBM que se muestran son precios de distribuidor recomendados por IBM, corresponden al momento actual y están sujetos a cambios sin aviso previo. Los precios de los distribuidores pueden ser diferentes.

Esta información sólo se proporciona a efectos de planificación. La información incluida en este documento puede cambiar antes de que los productos descritos estén disponibles.

Esta información contiene ejemplos de datos e informes utilizados en operaciones comerciales diarias. Para ilustrarlos de la forma más completa posible, los ejemplos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier parecido con los nombres y las direcciones utilizados por una empresa real es mera coincidencia.

LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente, que ilustran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier forma sin realizar ningún pago a IBM, con el fin de desarrollar, utilizar, comercializar o distribuir programas de aplicación para la interfaz de programación de aplicaciones para la plataforma operativa para la que se han escrito los programas. Estos ejemplos no se han probado exhaustivamente bajo todas las condiciones. IBM, por lo tanto, no puede garantizar ni dar a entender la fiabilidad, utilidad o funcionamiento de estos programas. Los programas de ejemplo se proporcionan "TAL CUAL", sin garantía de ningún tipo. IBM no se responsabiliza de ningún daño resultante de la utilización de los programas de ejemplo.

Cada copia o parte de estos programas de ejemplo o cualquier trabajo derivado debe incluir un aviso de copyright como el siguiente:

Algunas partes de este código proceden de los programas de ejemplo de IBM Corp. Sample Programs.

© Copyright IBM Corp. _especifique el año o los años_. Todos los derechos reservados.

Marcas registradas

IBM, el logotipo de IBM e ibm.com son marcas registradas o marcas comerciales registradas de International Business Machines Corp., registrada en muchas jurisdicciones en todo el mundo. Otros nombres de productos y servicios pueden ser marcas registradas de IBM u otras compañías. En la web encontrará una lista actual de marcas registradas de IBM, en www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript y todas las marcas comerciales basadas en Adobe son marcas registradas de Adobe Systems Incorporated en Estados Unidos o en otros países.

IT Infrastructure Library es una marca registrada de la agencia Central Computer and Telecommunications Agency, que forma parte de la Office of Government Commerce británica.

Intel, el logotipo de Intel, Intel Inside, el logotipo de Intel Inside, Intel Centrino, el logotipo de Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium y Pentium son marcas registradas de Intel Corporation o sus subsidiarias en Estados Unidos y en otros países.

Linux es una marca registrada de Linus Torvalds en Estados Unidos o en otros países.

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas registradas de Microsoft Corporation en Estados Unidos o en otros países.

ITIL es una marca registrada y una marca registrada de comunidad de Minister for the Cabinet Office, registrada en la Oficina de patentes y marcas registradas de EE.UU.

UNIX es una marca registrada de The Open Group en Estados Unidos y en otros países.

Java™ y todas las marcas registradas y logotipos basados en Java son marcas comerciales o marcas registradas de Oracle y/o sus filiales.

Cell Broadband Engine es una marca registrada de Sony Computer Entertainment, Inc. en Estados Unidos o en otros países, desde donde se utiliza bajo licencia.

Linear Tape-Open, LTO, el logotipo de LTO, Ultrium y el logotipo de Ultrium son marcas registradas de HP, IBM® Corp. y Quantum en Estados Unidos y en otros países.

Términos y condiciones de la documentación del producto

Se otorga permiso para el uso de estas publicaciones si se cumplen estos términos y condiciones.

Validez

Estos términos y condiciones se añaden a los términos de uso del sitio web de IBM.

Utilización personal

Puede reproducir estas publicaciones para su uso personal, no comercial, siempre que se conserven todos los avisos sobre derechos de propiedad. No puede distribuir, mostrar o realizar trabajos derivados de estas publicaciones, ni ninguna de sus partes, sin el consentimiento expreso de IBM.

Uso comercial

Puede reproducir, distribuir y visualizar estas publicaciones únicamente en su empresa siempre que se conserven todos los avisos de propiedad. No puede realizar ninguna tarea que derive de estas publicaciones ni reproducir, distribuir o visualizar estas publicaciones o una parte de las mismas fuera de su empresa, sin el consentimiento explícito de IBM.

Derechos

A excepción de lo especificado expresamente en este permiso, no se concede ningún otro permiso, licencia o derecho, ni explícito ni implícito, para la información o los datos, el software ni ninguna otra propiedad intelectual que contenga.

IBM se reserva el derecho de retirar los permisos que se hayan proporcionado siempre que, bajo su discreción, el uso de las publicaciones sea perjudicial para sus intereses o, según determine IBM, no se estén siguiendo adecuadamente las instrucciones detalladas anteriormente.

No se puede descargar, exportar o reexportar si no es en total cumplimiento con todas las leyes y reglamentos aplicables, incluidas las leyes y reglamentos de los EE.UU. en materia de exportación.

IBM NO PROPORCIONA NINGUNA GARANTÍA SOBRE EL CONTENIDO DE ESTAS PUBLICACIONES. LAS PUBLICACIONES SE PROPORCIONAN "TAL CUAL" Y SIN NINGÚN TIPO DE GARANTÍA, NI EXPLÍCITA NI IMPLÍCITA, INCLUIDAS, AUNQUE SIN LIMITARSE A, LAS GARANTÍAS DE COMERCIALIZACIÓN, NO CONTRAVENCIÓN Y ADECUACIÓN A UN PROPÓSITO DETERMINADO.



Impreso en España