

IBM Endpoint Manager  
Versión 9.2

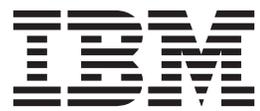
*Guía del lenguaje de relevancia*

**IBM**



IBM Endpoint Manager  
Versión 9.2

*Guía del lenguaje de relevancia*



**Aviso**

Antes de utilizar esta información y el producto al que da soporte, lea la información del apartado "Avisos" en la página 69.

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 2013, 2014.

---

# Contenido

## Capítulo 1. Introducción al lenguaje de relevancia . . . . . 1

Ámbito de la relevancia . . . . .	2
Utilización del depurador de Fixlet . . . . .	5
Utilización del depurador de presentación . . . . .	6

## Capítulo 2. Utilización de la relevancia . . . . . 9

Visión general del lenguaje de relevancia . . . . .	9
Elementos primarios . . . . .	10
Exists . . . . .	11
Plurales (recopilaciones) . . . . .	12
Whose - It . . . . .	13
Conjuntos . . . . .	17
Propiedades y referencias. . . . .	18
Relaciones . . . . .	18
Conversión . . . . .	19
Indexación. . . . .	20
Tuplas . . . . .	20
Plurales con tuplas . . . . .	21
Comparación de Tuplas . . . . .	22
Aritmética . . . . .	23
AND y OR . . . . .	23
If-then-else . . . . .	24
Expresiones . . . . .	25
Inspectores de IBM Endpoint Manager . . . . .	25
Inspectores básicos . . . . .	26
Otros ejemplos del inspector. . . . .	26
Propagación de no existencia . . . . .	27
Determinación de las propiedades del objeto . . . . .	27
Relevancia en el análisis de propiedades. . . . .	28
Visualización de análisis de propiedades . . . . .	28
Creación de análisis de propiedades . . . . .	29
Relevancia en los scripts de la acción. . . . .	29
Visualización de scripts de acción . . . . .	29
Creación de scripts de acción . . . . .	30
Expresiones de relevancia útiles . . . . .	31

Utilización de rangos de números . . . . .	31
Utilización del operador de barra . . . . .	33
Manipulación de cadenas. . . . .	33
Índice de subcadena de cadena de caracteres . . . . .	34
Manipulación de fechas y horas . . . . .	35
Suma de unidades de tiempo . . . . .	37
Operación en listas de números de coma flotante . . . . .	37
Utilización de búsquedas de comodín para archivos y carpetas . . . . .	38
Comparación de versiones . . . . .	38
Inspección del registro de Windows . . . . .	39
Variables de entorno . . . . .	41
Determinación de la información del sistema operativo . . . . .	42
Acceso al Administrador de tareas (Procesos) . . . . .	43
Examen de tareas planificadas y de ejecución . . . . .	44
Reconocimiento de Service Packs de Office . . . . .	44
Detección de Service Packs de idioma extranjero . . . . .	45
Deconstrucción de XML . . . . .	46
Utilización de listas blancas . . . . .	47

## Apéndice A. Gramática de lenguaje de relevancia . . . . . 49

Operadores de relevancia. . . . .	50
Precedencia y asociatividad . . . . .	50
Frases de clave de relevancia . . . . .	51
Historial del lenguaje . . . . .	54
BES 1.x . . . . .	54
Mensajes de error . . . . .	56

## Apéndice B. Glosario . . . . . 65

## Apéndice C. Soporte . . . . . 67

## Avisos . . . . . 69



---

## Capítulo 1. Introducción al lenguaje de relevancia

**IBM Endpoint Manager** permite supervisar y aplicar parches fácilmente a grandes redes de sistemas en tiempo real utilizando la tecnología **Fixlet**. Un Fixlet inspecciona un sistema cliente e informa a los servidores centrales. Esto permite que se aplique de forma eficaz un parche o una actualización a sólo aquellos equipos donde sea relevante, y no a otros. También permite la recuperación de varias propiedades del equipo que se puedan recopilar, analizar, representar gráficamente y archivar.

El núcleo de la tecnología Fixlet es el **Lenguaje de relevancia**, que permite a los autores interrogar las propiedades de hardware y software de los clientes gestionados utilizando **Inspectores**. Con el lenguaje de relevancia, puede grabar expresiones que describen casi cualquier aspecto del entorno del cliente. Algunos Fixlets están diseñados de forma sencilla para devolver información de relevancia en los servidores, pero la mayoría de ellos sugieren **acciones** que pueden aplicar parches o actualizar el equipo del cliente. Las acciones, a su vez, también se benefician de las expresiones de relevancia.

Los mensajes del fixlet y las expresiones de relevancia por ellas mismas sólo pueden *notificar* al usuario o al administrador. Las acciones, por otra parte, están específicamente diseñadas para *modificar* el cliente, por lo que hay una clara línea divisoria entre una expresión de relevancia y su acción asociada – normalmente, se necesita un humano para que despliegue la acción.

Al dividir la labor de esta manera, utilizando las cláusulas de relevancia para inspeccionar benignamente el cliente y las acciones para repararlas (tras su aprobación), las aplicaciones de IBM Endpoint Manager proporcionan una mezcla sin precedentes de seguridad y potencia.

Las expresiones de relevancia están diseñadas para ser legibles. Esto permite a los usuarios y a los administradores examinarlas antes de desplegar cualquier acción asociada. El lenguaje ofrece acceso a miles de propiedades de equipo mediante inspectores. Los valores devueltos por los inspectores se pueden utilizar para cálculos y comparaciones, lo que permite al Fixlet determinar la relevancia y marcar como objetivo un equipo para acción.

Esta guía es la que le servirá de referencia para el lenguaje de relevancia. Con esta guía y las *Guías de inspectores* específicas de la plataforma adecuadas, podrá grabar expresiones de relevancia que pueden desencadenar mensajes del Fixlet. Puede obtener más información sobre cómo crear acciones leyendo la *Referencia del lenguaje de acciones de IBM Endpoint Manager*.

Esta referencia es para gestores de TI que deseen grabar mensajes de Fixlet para sistemas gestionados por IBM Endpoint Manager. El paquete estándar de IBM Endpoint Manager incluye decenas de miles de Fixlets preempaquetados y de acciones asociadas, pero puede ampliar estas ofertas con contenido personalizado ajustado a su configuración de seguridad y de empresa específica.

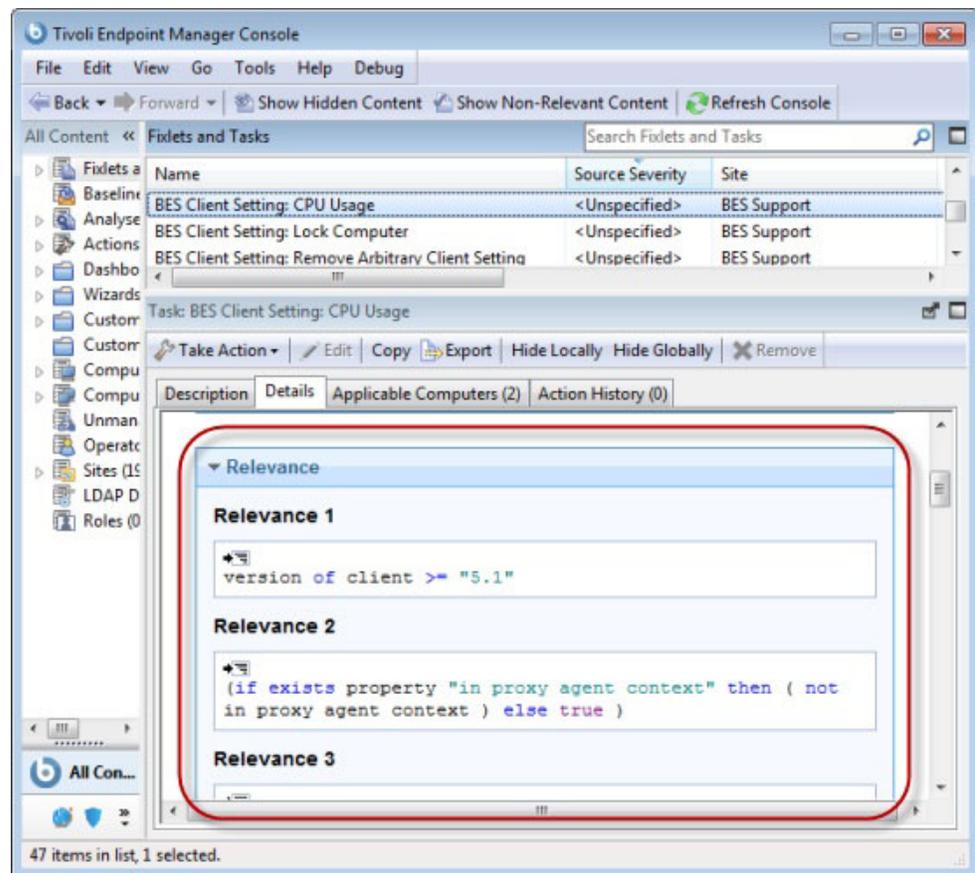
Esta referencia es también para gestores de TI que deseen comprender mejor las expresiones de relevancia que desencadenan un mensaje de Fixlet. Aunque el lenguaje está diseñado para ser legible, es posible realizar expresiones sofisticadas

que requieran de un cuidadoso análisis. Esto permitirá que los usuarios comprendan las cláusulas de relevancia incluidas también en los scripts de acción.

## Ámbito de la relevancia

La relevancia es una hebra continua que se utiliza en todos los aspectos del IBM Endpoint Manager. Algunas de sus manifestaciones más importantes incluyen:

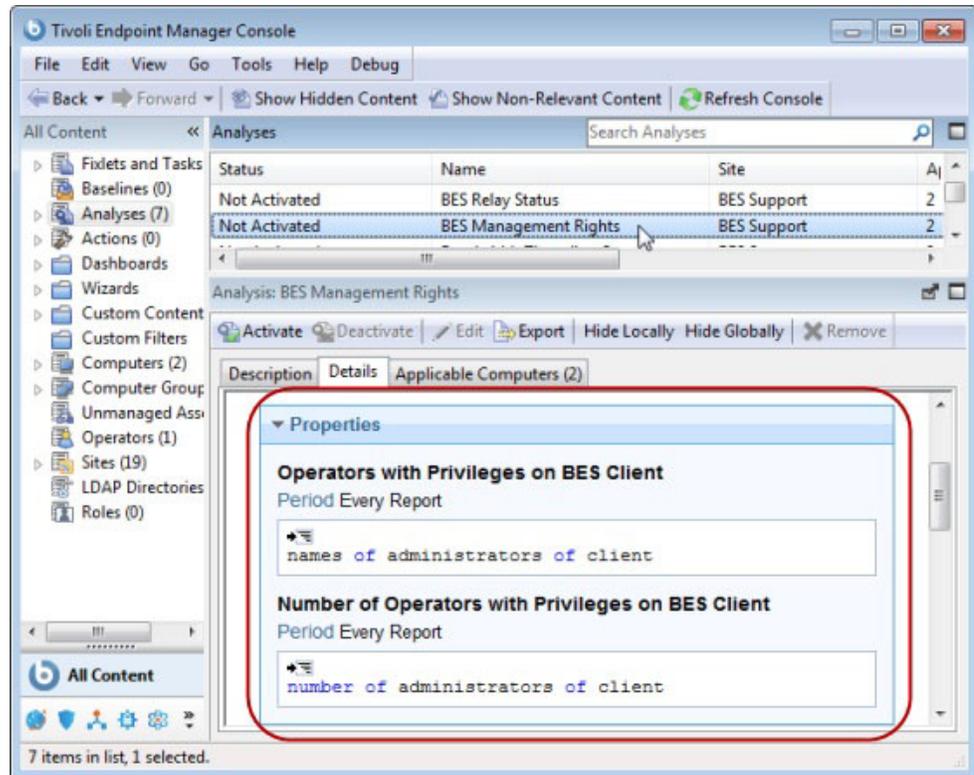
- **Evaluación de la relevancia del Fixlet.** Este es el lugar más común para buscar el lenguaje de relevancia y explica cómo ha obtenido su nombre. Como verá en el detalle inferior, las expresiones de este lenguaje están diseñadas para desencadenarse únicamente cuando el equipo cliente presente un estado determinado, por lo tanto el Fixlet no se visualizará a menos que sea relevante. Puede ver la expresión de relevancia detrás de cada Fixlet en la consola de IBM Endpoint Manager: pulse en un **Fixlet** en la lista y, a continuación, consulte el separador **Detalles**.



Puede ver más de una expresión de relevancia; están unidas mediante AND para formar la expresión final.

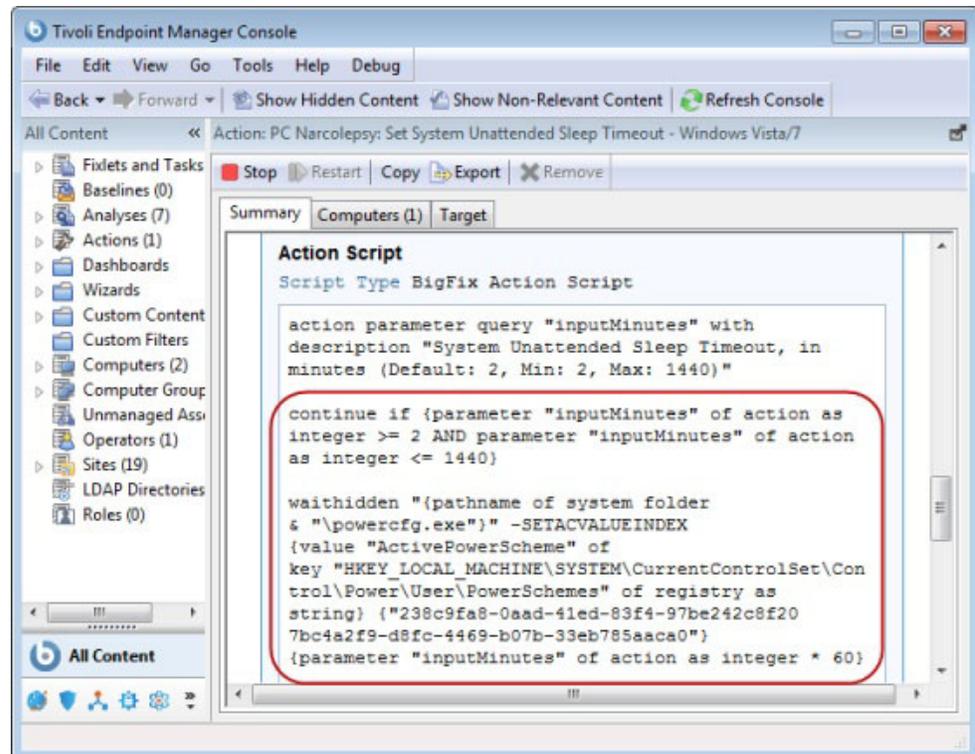
- **Evaluación de tareas.** Las tareas son funcionalmente similares a los Fixlets y utilizan las expresiones de relevancia para determinar a qué sistemas se deben dirigir. Siempre que los Fixlets se diseñen teniendo en cuenta la recuperación, las tareas están diseñadas teniendo en cuenta el mantenimiento de continuidad. La principal diferencia entre los Fixlets y las tareas recae en cómo se juzgan para ser "arreglados". Las acciones del Fixlet se mostrarán como corregidas cuando ya no sean relevantes, mientras que de forma predeterminada las acciones de tareas se mostrarán como corregidas una vez que todas las líneas de su acción se hayan completado.

- **Visualización de las propiedades recuperadas.** Desde la consola de IBM Endpoint Manager, puede recuperar propiedades de los sistemas cliente. Hay algunas propiedades integradas, pero también puede crear las suyas propias. Para ver un ejemplo, seleccione un elemento del separador **Análisis** y, a continuación, pulse en el siguiente separador **Detalles**.



Aquí encontrará expresiones de relevancia denominadas que se utilizan para interrogar alguna propiedad del sistema cliente y para devolver un valor. Al utilizar la extensa biblioteca del inspector, puede crear sus propias expresiones de relevancia personalizadas para examinar propiedades como 'administradores de cliente', 'fabricante del sistema', 'marca de cpu', 'Servidores DNS', 'sistema operativo' y cientos más.

- **Utilización de relevancia en acciones.** Las acciones tienen su propio lenguaje, pero pueden incorporar cláusulas de relevancia que se evalúan en tiempo de ejecución. Ello significa que el mismo conjunto de inspectores potente que puede utilizar para marcar como objetivo un cliente también se puede utilizar para personalizar la acción.



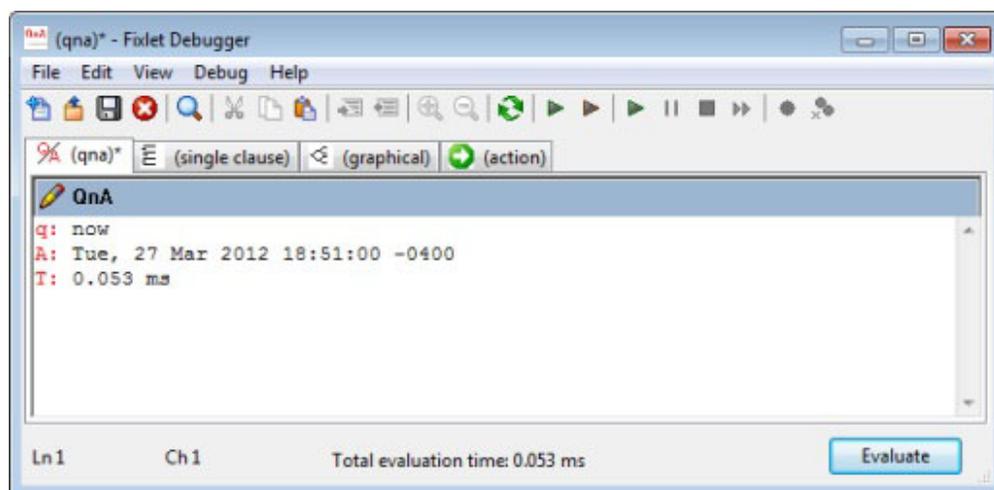
En este contexto, las cláusulas de relevancia se escriben entre llaves, como {parameter "inputMinutes" ...} y {pathname of system folder...}. Las cláusulas de relevancia booleanas se pueden utilizar para controlar las sentencias **if** y, de lo contrario, controlar el flujo del script de acción. Se pueden utilizar como aserciones durante el tiempo de ejecución sobre la validez de algún procedimiento o dato. Se utiliza generalmente en acciones del Fixlet para asegurarse de que un archivo descargado tenga el tamaño adecuado y un valor de hash antes de continuar. Las aserciones hacen el código más seguro, más sólido y más fácil de depurar.

- **Creación de informes en el despliegue de IBM Endpoint Manager.** Los inspectores de la sesión se pueden utilizar para visualizar el estado del propio despliegue de IBM Endpoint Manager. Hay centenares de inspectores que pueden examinar Fixlets, acciones, sistemas, usuarios, propiedades, asistentes, etc. También se proporciona un amplio conjunto de medidas estadísticas para ayudarle a analizar, informar y representar gráficamente el estado de su despliegue.

En las siguientes secciones, presentamos dos de las principales herramientas para experimentar, probar y depurar sus propias expresiones de relevancia. Tenga en cuenta que hay dos ubicaciones distintas para las expresiones de relevancia: cliente y sesión. La relevancia del cliente le permite inspeccionar y reparar los puntos finales de la red. La relevancia de sesión le permite analizar la base de datos central. Estos dos grupos no siempre se solapan, y ver expresiones de relevancia en el depurador incorrecto puede proporcionar resultados incorrectos.

## Utilización del depurador de Fixlet

Para probar y depurar la relevancia de los clientes, también hay un depurador autónomo llamado depurador de Fixlet (anteriormente denominado depurador de relevancia o QnA) que puede utilizar.



A continuación se muestra cómo utilizarlo:

1. Inicie la sesión como administrador y ejecute el programa **FixletDebugger.exe**. Se encuentra en **Archivos de programa > BigFix Enterprise > Consola de BES > QnA**.
2. Pulse **Sí** para permitir que el programa se ejecute.
3. Escriba una expresión de relevancia precedida por **Q:**, como por ejemplo  
**Q: now**
4. Pulse el botón **Evaluar**.
5. Debajo de la expresión, verá la evaluación, como  
**Q: now**  
**A: Tue, 27 Mar 2012 18:51:00 -0400**  
**T: 0.053 ms**

Tenga en cuenta que, además de que devuelve la respuesta, el programa también puede visualizar el tiempo (en milisegundos) que se tarda en procesar la solicitud. Es importante al intentar evitar las evaluaciones que requieren mucho tiempo. Para mostrar la hora, selecciónela como una opción desde el menú **Ver**.

Si se detecta un error, se imprimirá un mensaje precedido de 'E:'. (Consulte el Apéndice para ver descripciones de los mensajes de error de relevancia).

Hay opciones en el menú **Ver** que proporcionan más información:

**Mostrar hora de evaluación:** para analizar el rendimiento, seleccione este valor. Le mostrará el tiempo transcurrido de la ejecución de relevancia en microsegundos. Es importante para crear mensajes de Fixlet que tengan tanta capacidad de respuesta como sea posible.

**Mostrar información de tipo:** puede ver el tipo de inspector del objeto devuelto seleccionando esta opción. El examen del tipo devuelto le ayudará a saber cómo combinar correctamente los resultados con expresiones más complicadas.

Esta guía presenta muchos ejemplos en el formato QnA (pregunta y respuesta), para facilitar la continuación. Los ejemplos se encuentran en una font Courier, precedidos por una viñeta cuadrada roja. Por ejemplo:

```
Q: names of files of folder "c:/"
A: AUTOEXEC.BAT
A: boot.ini
A: CONFIG.SYS
A: IO.SYS
A: MSDOS.SYS ...
```

```
T: 1.944 ms
I: plural string
```

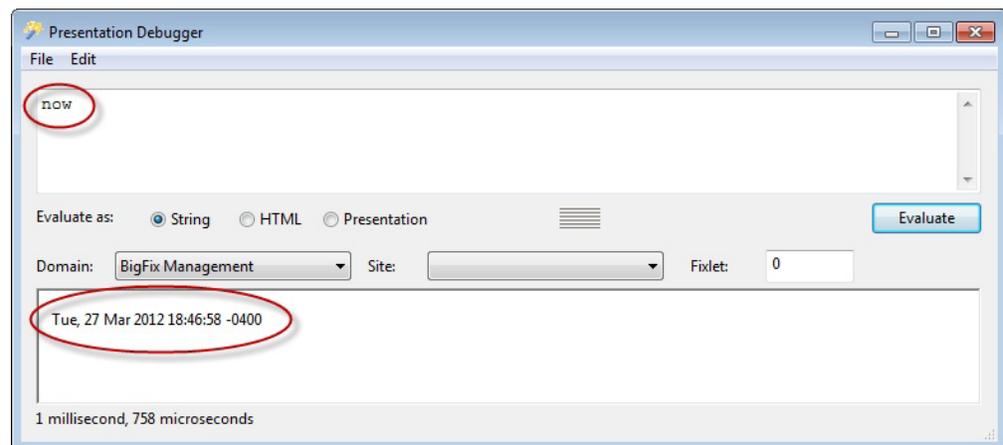
Este fragmento de relevancia devuelve los nombres de los archivos de la unidad C: (muestra una lista parcial), cada una precedida por una 'A:'. El tiempo para recuperar esta información es 1.944 microsegundos y el tipo de devolución es una cadena de caracteres plural.

---

## Utilización del depurador de presentación

IBM Endpoint Manager incluye herramientas para ayudarle a grabar y depurar expresiones de relevancia de la sesión. Siga estos pasos para instalar el depurador de la sesión (también denominada presentación):

1. Mientras que la consola de IBM Endpoint Manager está en ejecución, pulse **Ctrl-Mayús-Alt-D** para abrir la ventana **Depurar**.
2. Pulse el recuadro de selección situado junto a **Mostrar menú de depuración**, en la parte superior de la ventana. Esto instala un nuevo menú en la Consola denominado **Depurar** que contiene varias herramientas de depuración útiles.
3. En el menú de depuración, pulse **Depurador de presentación** para abrir la ventana Depurador de presentación.



Para utilizar el depurador:

1. Escriba una expresión de relevancia en el cuadro superior. Como un ejemplo sencillo, escriba la frase clave **now**. Esta expresión extrae la fecha y la hora actuales del reloj del sistema.

2. Pulse el botón **Evaluar**.
3. En el cuadro de texto inferior, se mostrará la fecha actual.



---

## Capítulo 2. Utilización de la relevancia

---

### Visión general del lenguaje de relevancia

El lenguaje de relevancia, junto con las extensiones del inspector, está diseñado para permitir que extraiga de los equipos del cliente información útil, o para ver si necesitan recuperación. Los inspectores son las frases principales del lenguaje de relevancia, de modo que veamos cómo se forman. A continuación mostramos una página ilustrativa de la Guía de inspectores de Windows:

#### Operating System

The operating system object provides access to several important properties of the system.

#### Creation Methods

Key Phrase	Form	Description
operating system	<i>PlainGlobal</i>	Creates the global operating system object. <small>Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu</small>

#### Properties

Key Phrase	Form	Return Type	Description
<operating system> as string	<i>Cast</i>	<string>	Returns a string containing the name of the operating system concatenated with the release. <small>Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu</small>
boot time of <operating system>	<i>Plain</i>	<time>	Returns the time of the last restart. <small>Win, Lin, Sol, HPUX, AIX, Mac, Ubu</small>
build number high of <operating system>	<i>Plain</i>	<integer>	Numeric representation of the most significant 16 bits of the build number. <small>Win, WM</small>
build number low of <operating system>	<i>Plain</i>	<integer>	Numeric representation of the least significant 16 bits of the build number. <small>Win, WM</small>
csd version of <operating system>	<i>Plain</i>	<string>	Returns the Corrective Service Disk version of the operating system. The szCSDVersion as returned by the GetVersionEx system call. The format varies depending on the installed service packs. For WinNT it contains a string such as "Service Pack 3", for Win95 it can contain a string such as "B". <small>Win, WM</small>
ia64 of <operating system>	<i>Plain</i>	<boolean>	Returns TRUE iff the BES Client is running on Itanium. <small>Win7.0, WM</small>
mac of <operating system>	<i>Plain</i>	<boolean>	Returns TRUE if the client computer is a Macintosh. <small>Win8.0, Lin8.0, Sol8.0, HPUX8.0, AIX8.0, Mac8.0, Ubu</small>

Un Inspector tiene uno o varios métodos de creación para definir el objeto. Cada objeto definido, a su vez, tiene propiedades que se pueden inspeccionar. Aquí, el sistema operativo es el nombre del objeto y las propiedades incluyen varios aspectos del sistema operativo, entre los que se incluyen los números de compilación y los tiempos de arranque.

Para ilustrar los inspectores y los elementos de relevancia específicos, las siguientes secciones incluyen ejemplos que utilizan el depurador de Fixlet (utilizando el estilo de visualización de QnA). Si es posible, ejecute el programa (FixletDebugger.exe) y especifique los ejemplos a medida que lo ejecute.

## Elementos primarios

Los bloques de creación básicos del lenguaje son números, series y expresiones que los combinan.

```
Q: "hello world"  
A: hello world
```

Este ejemplo resalta una cadena de caracteres.

Las cadenas de literales como esta se analizan para un carácter especial: el signo de porcentaje. Es un carácter de escape que se codifica para otros caracteres, incluidos caracteres de control y suprimir. Cuando se encuentra un signo de porcentaje, la codificación espera que los dos siguientes caracteres sean dígitos hexadecimales que creen un valor hexadecimal de un solo byte. El valor hexadecimal se añadirá a la representación interna de la cadena de caracteres, lo que le permite incorporar caracteres que de otra forma no estarían disponibles en una cadena de caracteres. Como el signo de porcentaje se utiliza como la tecla de escape, para obtener un porcentaje en una cadena de caracteres debe utilizar %25, el valor hexadecimal de porcentaje.

Las cadenas de caracteres no son las únicas primitivas:

```
Q: 6000  
A: 6000
```

Este ejemplo anterior prueba un entero. También puede hacer matemáticas:

```
Q: (8+3)*6  
A: 66
```

Los elementos primarios incluyen expresiones entre paréntesis como la (8+3) anterior. Estos elementos primarios también se pueden desglosar:

```
Q: substrings separated by "-" of "an-over-hyphenated-string"  
A: an  
A: over  
A: hyphenated  
A: string  
I: plural substring
```

Tenga en cuenta que en el ejemplo anterior se han devuelto los cuatro valores, no sólo uno. Este resultado es habitual de un inspector de plural como 'substrings'. Puede filtrar esta lista con una sentencia 'whose':

```
Q: (substrings separated by " " of "who observed what happened, when and where?")  
whose (it contains "w")  
A: who  
A: what  
A: when  
A: where?  
I: plural substring
```

Este ejemplo muestra dos cláusulas entre paréntesis. La primera cláusula entre paréntesis crea una lista de palabras (subcadenas separadas por un espacio). Esta cláusula 'whose' contiene la palabra clave primaria 'it' (de la que se hablará con más detalle posteriormente), que se puede sustituir por otro objeto – en este caso,

'it' se sustituye por cada una de las palabras individuales, y la expresión devuelve aquellas palabras que contienen la letra 'w'. ¿Cuántas de estas cadenas de caracteres hay?

Q: number of (substrings separated by " " of "who observed what happened, when and where?")  
whose (it contains "w")  
A: 4

Esta expresión muestra cómo puede realizar el recuento del número de elementos devueltos y filtrados desde un inspector plural. Como muestran estos ejemplos, puede obtener elementos singulares o plurales de una expresión de relevancia. ¿Y si no hay ningún elemento? Ese es un asunto para la siguiente sección.

## Exists

**Exists** es una palabra clave importante que devuelve TRUE o FALSE en función de la existencia del objeto especificado. Se trata de una técnica importante que le permite probar la existencia antes de probar un valor y, posiblemente, incurrir en un error. La palabra clave tiene dos usos típicos ligeramente distintos. El primero es para determinar si un objeto *singular* especificado por un inspector existe:

Q: exists drive "c:"  
A: True  
Q: exists drive "z:"  
A: False

Los ejemplos anteriores prueban la existencia de los objetos especificados en el sistema del cliente. En estos ejemplos, puede ver que el cliente tiene una unidad c:, pero no una unidad z:. Si intenta buscar más información sobre la unidad no existente, puede generar un error. Si no está seguro de la existencia de un objeto, utilice la palabra clave 'exist' antes de intentar examinar sus propiedades.

El segundo uso es determinar si un resultado *plural* contiene valores:

Q: exists (files of folder "c:")  
A: True

Esta expresión devuelve TRUE, ya que existen los archivos en la unidad c:. Tenga en cuenta que la utilización de la propiedad plural (archivos) es una forma segura de hacer referencia a algo que puede existir o no. Por ejemplo:

Q: file of folders "z:"  
E: Singular expression refers to nonexistent object.

Se genera un error aquí porque no hay ninguna unidad "z:" en el sistema del cliente. Si solicita una respuesta plural,

Q: files of folders "z:"  
I: plural file

No se le otorga una respuesta (no es una: contestación), pero tampoco arroja un error. No obstante, ambas construcciones se pueden examinar con la palabra clave 'exists' sin provocar un error:

Q: exists file of folders "z:"  
A: False  
Q: exists files of folders "z:"  
A: False

## Plurales (recopilaciones)

Como se vio en la sección anterior, los plurales de los inspectores son sencillos de crear, sólo añadiendo una 's' al final del nombre. 'Substring' es singular, 'substrings' es plural:

```
Q: substrings separated by " " of "a short string"
A: a
A: short
A: string
I: plural substring
```

Pero un inspector plural no tiene que devolver un resultado plural:

```
Q: substrings separated by " " whose (it contains "o") of "a short string"
A: short
I: plural substring
```

Aunque el resultado es un tipo de subcadena plural, sólo hay un solo valor. De hecho, como se vio en la última sección, una expresión plural puede que no devuelva ningún valor en absoluto, sin incurrir en un error:

```
Q: substrings separated by " " whose (it contains "z") of "a short string"
I: plural substring
```

Esto no devuelve valores, pero tampoco ningún error. Por lo tanto, es importante recordar que la pluralidad es una propiedad de la propia expresión, no necesariamente los resultados.

Además, hay restricciones en expresiones singulares. Cuando un plural pueda devolver cero, uno o más valores, se espera que una expresión singular devuelva exactamente un valor. Por ejemplo,

```
Q: substring separated by " " whose (it contains "o") of "a short string"
A: short
I: singular substring
```

Debería estar esperando un valor solitario como este como resultado de la evaluación de un inspector singular. Sin embargo, lo siguiente devuelve un error:

```
Q: substring separated by " " whose (it contains "s") of "a short string"
E: Singular expression refers to non-unique object.
```

Esto se debe a que hay dos palabras que contienen 's', y esta expresión busca un valor singular. Mientras que dos es demasiado, cero no es suficiente:

```
Q: substring separated by " " whose (it contains "z") of "a short string"
E: Singular expression refers to nonexistent object.
```

Si está seguro de recuperar un valor solitario, utilice la versión singular. De lo contrario, para una mayor flexibilidad, utilice el plural. Como ejemplo práctico, puede encontrar una carpeta única como esta:

```
Q: name of folder of folder "c:/Documents and Settings"
A: All Users
E: Singular expression refers to non-unique object.
```

Pero como podrá ver, aunque devuelve una respuesta, también genera un error. Se debe a que hay varias carpetas en la ubicación especificada, y este mandato sólo recupera la primera. Para ver la lista completa, debe utilizar la versión plural:

Q: names of folders of folder "c:/Documents and Settings"  
A: All Users  
A: Default User  
A: LocalService  
A: NetworkService ...

Puede crear explícitamente plurales utilizando un punto y coma (;) para separar los elementos. Estos se denominan recopilaciones:

Q: "two"; "words"  
A: two  
A: words  
Q: exist files ("c:\whitelist.txt"; "c:\blacklist.txt")  
A: True  
Q: conjunction of (True; True)  
A: True  
Q: conjunction of (True; False)  
A: False

Las últimas dos expresiones de relevancia AND junto con el punto y coma separaban la recopilación. Tenga en cuenta que los plurales deben ser del mismo tipo, o generará un error:

Q: "one"; 1  
E: Incompatible types.

Si desea combinar distintos tipos, utilice una tupla (véase más abajo).

## Whose - It

'Whose' e 'it' son un par popular en el lenguaje de relevancia, aunque 'it' tiene una vida propia. En las siguientes secciones se detalla en primer lugar 'whose' y, a continuación, 'it', pero de necesidad, hay mucho solapamiento.

### Whose

La cláusula '**whose**' le permite filtrar un resultado o un conjunto de resultados según los criterios de relevancia especificados. Tiene este formato:

<list> whose <filter expression>

Por ejemplo:

Q: (1;2;3;5;8;17) whose ( it mod 2 = 1 )  
A: 1  
A: 3  
A: 5  
A: 17

La palabra clave especial '**it**' hace referencia a los elementos de la lista (en este caso, la recopilación de números) y se vincula únicamente en la expresión de filtro. El lenguaje de relevancia ejecuta la expresión de filtro una vez para cada valor de la propiedad filtrada, con 'it' haciendo referencia, a su vez, a cada resultado. Los resultados donde la cláusula de filtro se evalúa en TRUE se incluyen en la lista de salida. Tenga en cuenta que 'it' hace referencia a la lista inmediatamente a la izquierda de la sentencia 'whose' (fuera de una sentencia 'whose', 'it' hace referencia al primer elemento a la derecha de los paréntesis).

'It' también puede hacer referencia a objetos directos que no forman parte de una cláusula whose:

Q: (it \* 2) of (1;2;3)  
A: 2  
A: 4  
A: 6

Aquí, 'it' toma los valores de la lista, uno cada vez.

También puede utilizar paréntesis para definir el ámbito de los objetos whose-it. Un uso racional de los paréntesis puede asegurar resultados adecuados mientras que mejora la legibilidad. Por ejemplo, los ejemplos siguientes muestran cómo una pequeña redistribución de cláusulas whose puede cambiar el resultado de forma significativa:

Q: preceding texts of characters of "banana" whose (it contains "n")  
A:  
A: b  
A: ba  
A: ban  
A: bana  
A: banan

Q: preceding texts of characters of ("banana" whose (it contains "n"))  
A:  
A: b  
A: ba  
A: ban  
A: bana  
A: banan

Estas expresiones van carácter por carácter a través de la palabra 'banana' y devuelven el texto anterior a cada carácter. Dado que devuelve el texto antes del carácter, devuelve el espacio en blanco antes de 'b' y se detiene en la 'a' final con 'banan'. Las expresiones devuelven los mismos valores, pero el segundo clarifica más a qué hace referencia 'it', es decir 'banana'. Como 'banana' siempre tendrá una 'n', esta expresión devolverá *todas* las subcadenas especificadas.

Q: preceding texts of characters whose (it contains "n") of "banana"  
A: ba  
A: bana

Q: preceding texts of (characters of "banana") whose (it contains "n")  
A: ba  
A: bana

Estas dos expresiones son equivalentes, pero la segunda muestra más explícitamente a qué se refiere 'it', es decir, los caracteres de la palabra 'banana'. La 'n' aparece dos veces en banana, por lo que se devolverán las dos subcadenas.

Q: preceding texts whose (it contains "n") of characters of "banana"  
A: ban  
A: bana  
A: banan

Q: (preceding texts of characters of "banana") whose (it contains "n")  
A: ban  
A: bana  
A: banan

Estas dos expresiones hacen lo mismo, y devuelven las subcadenas iniciales de 'banana' que contienen una 'n'.

En el uso práctico, podría utilizar cláusulas 'whose-it' para filtrar carpetas:

Q: names whose (it contains "a") of files of folder "c:"  
A: at170.dll  
A: blacklist.txt  
A: pagefile.sys...

O puede colocar la cláusula 'whose' al final de la expresión, lo que hace que el objeto de 'it' sea más explícito y sea más fácil de leer:

Q: (names of files of folder "c:") whose (it contains "a")  
A: at170.dll  
A: blacklist.txt  
A: pagefile.sys

Si la propiedad filtrada es singular, el resultado de la cláusula 'whose' es singular. Si la propiedad filtrada es un tipo plural, el resultado es un tipo plural.

Q: exists active device whose (class of it = "Display")  
A: True

Esta propiedad singular da un resultado true si hay un dispositivo de visualización activo en el sistema cliente.

Q: files whose (name of it starts with "x") of system folder  
A: "xactsrv.dll" "5.1.2600.2180" "Downlevel API Server DLL" "5.1.2600.2180 (xpsp\_sp2\_rtm.040803-2158)" "Microsoft Corporation"  
A: "xcopy.exe" "5.1.2600.2180" "Extended Copy Utility" "5.1.2600.2180 (xpsp\_sp2\_rtm.040803-2158)" "Microsoft Corporation"

Esta expresión plural devolverá una lista de archivos del sistema cuyos nombres empiezan con 'x'.

A medida que circula a través de los valores plurales, la expresión del filtro puede intentar evaluar un objeto no existente. Por sí mismo, tal expresión puede generar un error como:

E: Singular expression refers to nonexistent object.

Pero en el caso de una cláusula 'whose', el valor no existente simplemente se pasa por alto y se excluye del conjunto resultante. Como efecto secundario, esta característica le permite examinar un objeto en busca de su existencia antes de intentar inspeccionarlo (y generar un error). Como ejemplo, a continuación se muestra una cláusula de relevancia que desencadenará un error de existencia:

Q: exists file of folder "z:\bar"  
E: Singular expression refers to nonexistent object.

Pero, colocando esta cláusula dentro de una sentencia 'whose', puede evitar el error:

Q: exists folder "z:\bar" whose(exists files of it)  
A: False

## It

La palabra clave 'it' siempre hace referencia al objeto directo más próximo o al objeto de la cláusula 'whose' circundante más próxima, siempre que esté más cerca. Hay tres contextos simples en los que 'it' tiene un significado:

- <'it' expression> of <direct\_object>
- phrase (<'it' expression>) of <direct\_object>
- (<whose\_object>) whose ( <'it' expression> )

Los primeros dos contextos implican objetos directos, el tercero implica una cláusula 'whose'. Un ejemplo de un objeto directo es esta expresión, que lista los nombres y los tamaños de archivos de una carpeta especificada:

```
Q: (name of it, size of it) of files of folder "c:"
A: AUTOEXEC.BAT, 0
A: blacklist.txt, 42
A: boot.ini, 209
A: CONFIG.SYS, 0
...
A: whitelist.txt, 213
```

Aquí, 'it' hace referencia a los 'archivos de la carpeta "c:"'.

La cláusula 'whose' permite filtrar una lista basada en la evaluación de una expresión 'it'. Se trata de uno de los destinos más importantes de la palabra clave 'it':

```
Q: exist files whose (name of it starts with "b") of folder "c:"
A: True
Q: number of (files whose (name of it starts with "b") of folder "c:")
A: 2
```

En estas expresiones, 'it' sigue haciendo referencia a los 'archivos de la carpeta "c:"'.

Debe tener cuidado al colocar los paréntesis, ya que pueden cambiar el destino de la palabra clave 'it'. En la siguiente expresión, 'it' hace referencia a los archivos:

```
Q: (files of folder "c:") whose (name of it contains "a")
A: "atl70.dll" "7.0.9466.0" "ATL Module for Windows (Unicode)" "7.00.9466.0"
"Microsoft Corporation"
A: "blacklist.txt" "" "" "" ""
...
```

Tenga en cuenta que no es lo mismo que las siguientes expresiones de relevancia, que tienen una colocación incorrecta de paréntesis:

```
Q: files of folder "c:" whose (name of it contains "a")
E: Singular expression refers to nonexistent object.
Q: files of ( folder "c:" whose (name of it contains "a") )
E: Singular expression refers to nonexistent object.
```

Estas dos son sentencias equivalentes (e incorrectas) donde 'it' hace referencia al objeto más cercano, que es la carpeta, no los archivos.

Puede haber más de un 'it' en una expresión. La regla es que cada uno haga referencia a los objetos listados a la izquierda del 'whose' asociado. Por ejemplo:

```
Q: preceding texts whose (it contains "n") of characters whose (it is "a")
of "banana"
A: ban
A: banan
```

En este caso, la expresión devolverá las subcadenas anteriores 'a' que contienen 'n'. El primer 'it' hace referencia a las subcadenas; el segundo hace referencia a los caracteres. Esta regla sencilla e intuitiva facilita el desarrollo de expresiones complejas. A continuación se muestra otro ejemplo:

```
Q: (characters of "banana") whose (exists character whose (it is "n")
of preceding text
of it)
A: a
A: n
A: a
```

Esta expresión ilustra dos cláusulas whose-it anidadas. La interna busca subcadenas iniciales que contengan una 'n'. La externa devuelve los caracteres seguidos de dichas subcadenas.

Como 'it' representa un valor, puede operar en él como en cualquier otra variable:

```
Q: (it * it) of (1;2;3;4)
A: 1
A: 4
A: 9
A: 16
```

Puede anidar estas referencias:

```
Q: (it * it) whose (it > 8) of (1;2;3;4)
A: 9
A: 16
```

Aquí, las primeras instancias de 'it' se multiplican y se pasan a la tercera instancia de 'it' para su comparación.

'It' siempre hace referencia a un solo valor, y nunca a una lista.

## Conjuntos

Puede convertir una lista devuelta por un inspector plural en un conjunto matemático. Como tal, puede realizar operaciones de conjunto típicas como unión e intersección. Puede crear conjuntos de elementos individuales, separados por punto y coma:

```
intersection of (set of ("to";"be"); set of ("or";"not";"to";"be"))
```

Esta frase devuelve el conjunto compuesto de los dos elementos: be y to. Los conjuntos no se pueden representar directamente en el depurador. Para ver los elementos individuales en la lista, utilice el mandato elements:

```
Q: elements of intersection of (set of ("to";"be"); set of ("or";"not";
"to";"be"))
A: be
A: to
```

También puede crear conjuntos de listas normales. Aquí se muestra un ejemplo de utilización de inspectores de sesión que se deben ejecutar en el depurador de la presentación:

```
intersection of administered computer sets of bes users whose (name of it is "joe"
or name of it is "sue")
```

La frase "intersección" devuelve el conjunto de sistemas administrados por Sue y Joe. De forma similar, puede calcular la unión de conjuntos:

```
size of union of applicable computer sets of bes fixlets whose ((source severity
of it is "Critical") and (current date - source release date of it > 7 * day)) as
floating point / size of bes computer set as floating point
```

Esta expresión devuelve la proporción de sistemas que tienen al menos un Fixlet crítico relevante publicado hace más de una semana. Observe el uso de la palabra **tamaño** que devuelve el número de elementos del conjunto.

**Nota:** Las frases con "bes", como "bes fixlet" y "bes computer" son inspectores de **sesión** y sólo funcionarán en el depurador de presentación mientras la consola se encuentre en la sesión. Al intentar evaluar estas expresiones en el depurador del Fixlet se producirá un error que indica que el operador no está definido.

Establece también permitir resta:

set of (1;2;3;4) – set of (1;5)

Devuelve el conjunto formado por los elementos 2, 3 y 4. Tenga en cuenta que restar un número no en el conjunto original no afecta al resultado. Puede convertir el conjunto de nuevo a una lista imprimible, utilizando el mandato **elements**.

Q: elements of (set of (1;2;3;4) – set of (1;5))

A: 2

A: 3

A: 4

La palabra clave **elements** itera durante el objeto del conjunto, que devuelve los elementos del conjunto individuales como una lista normal.

## Propiedades y referencias

Se puede inspeccionar y hacer referencia a las propiedades de los objetos. Hay miles de inspectores de propiedades disponibles para cubrir la mayoría de las características de software y de hardware de los sistemas \*nix, Windows y Mac.

Q: day\_of\_week of current date

A: Tuesday

Devuelve una referencia al día de la semana desde la fecha del sistema de hoy.

Q: year of current date

A: 2012

Devuelve la parte del año de la fecha de hoy

Q: number of processors

A: 2

Devuelve el número de procesadores del sistema cliente.

Q: names of local groups

A: Administrators

A: Backup Operators

A: Guests

Devuelve una propiedad plural (names) como una lista correspondiente a los nombres de grupo locales.

Q: bit 0 of 5

A: True

Devuelve el bit cero (de orden inferior) como True (1) o False (0).

## Relaciones

Utilice relaciones para comparar valores en el lenguaje de relevancia. Son el estándar alfa y los comparadores numéricos (=, !=, <, >, >=, <=), así como algunas relaciones de cadenas de caracteres estrictamente (empieza con, termina con, contiene). A continuación se muestran algunos ejemplos de expresiones que utilizan relaciones:

Q: 1 < 2

A: True

Q: 2 is not less than or equal to 1

A: True

La cadena de caracteres compara el uso del orden alfabético:

Q: "the whole" is greater than "the sum of the parts"  
A: True

Algunas relaciones buscan subcadenas de otras cadenas:

Q: "nowhere" starts with "now"  
A: True  
Q: "nowhere" ends with "here"  
A: True  
Q: "nowhere" contains "her"  
A: True  
Q: "he" is contained by "nowhere"  
A: True

Las relaciones devuelven un booleano TRUE o FALSE en función del resultado de la comparación. A continuación se muestra una tabla de los símbolos de relación y de sus equivalentes en inglés:

Símbolo	Versión inglesa
=	es
!=	no es
<	es menor que
<=	es menor o igual que
>	es mayor que
>=	es mayor o igual que
	empieza con
	acaba en
	contiene
	se incluye en

## Conversión

Los tipos se pueden convertir, lo que simplifica la creación, la concatenación y la combinación de inspectores en expresiones de relevancia complejas.

Q: "01 Apr 2020" as date  
A: Wed, 01 Apr 2020

Convierte (casts) una cadena de caracteres en un tipo de fecha.

Q: 5 as month  
A: May

Convierte un entero en el tipo de mes correspondiente.

Q: january as three letters  
A: Jan

Convierte el mes de enero en una abreviación de tres letras.

El depurador del Fixlet convierte los valores en series para imprimirlos. Si un objeto no da como resultado una cadena de caracteres, el depurador utiliza el método 'as string' del objeto para convertirlo en una cadena de caracteres. Si el objeto no se puede convertir como una cadena de caracteres, se mostrará un mensaje de error.

## Indexación

Puede indexar en listas de objetos para seleccionar la propiedad deseada.

```
Q: line 2 of file "c:/frost_poem.txt"  
A: his house is in the village, though.
```

Devuelve la segunda línea del archivo de texto especificado.

```
Q: month 9  
A: September
```

Devuelve el nombre del noveno mes.

## Tuplas

Las tuplas añaden algunas propiedades útiles al lenguaje de relevancia. Una tupla es básicamente un tipo de compuesto formado de dos o más tipos distintos. Se puede devolver directamente desde un Inspector, como este:

```
Q: extrema of (1;2;3;4;5)  
A: 1, 5  
T: 0.127 ms  
I: singular ( integer, integer )
```

Esta cláusula de relevancia devuelve un objeto compuesto que incluye un rango de tiempo y un booleano asociado TRUE/FALSE. Tenga en cuenta el uso del operador de concatenación (&), que se utiliza aquí para crear un rango de tiempo (consulte los operadores aritméticos, a continuación).

Las tuplas también se pueden generar explícitamente mediante la utilización de la palabra clave coma (.). Se permite cualquier combinación de tipos:

```
Q: number of processors, "B or not", 8/4, character 66  
A: 2, B or not, 2, B  
I: ( integer, string, integer, string )
```

```
Q: now, "is the time"  
A: ( Fri, 22 Sep 2006 12:14:55 -0400 ), is the time  
I: ( time, string )
```

```
Q: 1, number of processors < 3, "friend"  
A: 1, True, friend  
I: ( integer, boolean, string )
```

Tenga en cuenta que si un Inspector individual devuelve una tupla, siempre devolverá los mismos tipos en el mismo orden. No es posible hacer que un Inspector devuelva tuplas de tipo <int, string, int> en un caso y <int, int, string> en otro.

Las tuplas se pueden indexar utilizando la palabra clave **'item'** (los índices comienzan en 0). Por ejemplo:

```
Q: item 0 of ("foo", 3, free space of drive of system folder)  
A: foo  
I: singular string
```

```
Q: (item 1 of it; item 2 of it) of ("foo", 3, free space of drive of system folder)  
A: 3  
A: 18105667584  
I: plural integer
```

Las tuplas proporcionan una forma en que la expresión de relevancia devuelve varias propiedades relacionadas. Por ejemplo, podría generar un conjunto de nombres de archivos y de tamaños de archivos correspondientes para todos los archivos que cumplan con un criterio específico con una sentencia de relevancia como esta:

```
Q: (name of it, size of it) of files whose ( size of it > 100000 ) of folder "c:"
A: hiberfil.sys, 536301568
A: ntldr, 250032
A: pagefile.sys, 805306368
I: plural ( string, integer )
```

## Plurales con tuplas

Las tuplas se pueden combinar con plurales para crear cláusulas de relevancia de sorprendente complejidad y potencia. La combinación más sencilla es también la menos útil. La formación de plurales de tuplas (del mismo tipo) crea una tupla de plural:

```
Q: (1,2); (3,4)
A: 1, 2
A: 3, 4
I: ( integer, integer )
```

Sin embargo, a la hora de intentar formar un plural de tuplas de *distintos* tipos, se genera un error. Como ya hemos visto, los plurales siempre deben ser del mismo tipo:

```
Q: (1,2);("a","b")
E: Incompatible types.
```

De forma interesante, la formación de una tupla de expresiones plurales genera un conjunto de tuplas que representan el producto de todos los plurales del componente:

```
Q: ((1; 2), ("a"; "b"), ("*"; "$" ))
A: 1, a, *
A: 1, a, $
A: 1, b, *
A: 1, b, $
A: 2, a, *
A: 2, a, $
A: 2, b, *
A: 2, b, $
I: plural ( integer, string, string )
```

Las tuplas de plurales también se pueden utilizar para buscar en dos listas afinidades. Por ejemplo, supongamos que tenemos dos listas de enteros, y deseamos saber qué números hay en la intersección de las listas. Podemos hacerlo utilizando una whose anidada y, a continuación, haremos referencia a la lista exterior envolviéndola en una tupla:

```
Q: (1;2;3;4) whose (exists (it, (2;4;6;8)) whose (item 0 of it is item 1 of it))
A: 2
A: 4
```

Lo negativo de este método es que la segunda lista está vinculada en la cláusula 'whose' y se debe volver a crear para cada iteración. Para mantener la capacidad de respuesta, debe mantener las listas tan cortas como se muestra a continuación.

Las tuplas de los plurales también se pueden utilizar para comparar dos conjuntos de datos:

```
Q:((1;2;3;4),(5;6;7;8)) whose (item 1 of it = 2*item 0 of it)
A: 3, 6
A: 4, 8
```

También puede averiguar qué archivos están en común comparando en serie las tuplas de 'new folder, old folder':

```
Q: (names of files of folder "c:/") whose (exists (it, (names of files of folder
"c:/old C"))) whose (item 0 of it is item 1 of it))
A: CONFIG.SYS
A: IO.SYS
A: MSDOS.SYS
A: report.txt
```

## Comparación de Tuplas

Puede comparar directamente tuplas con los operadores de comparación habituales, <, >, =. A continuación, se muestran algunos ejemplos:

```
Q: (1, 2, 3) < (2, 3, 4)
A: True
Q: ("a", "b", "c") < ("a", "b", "d")
A: True
```

La comparación continúa a través de los elementos de izquierda a derecha, y el primer par de elementos que falla la prueba finaliza la comparación de la tupla:

```
Q: ("abc", 17, 25, 4) < ("zzz", 17, 25, 4)
A: True
```

Esto es verdadero incluso aunque el resto de los números sean iguales, porque la prueba falla en la comparación del elemento más a la izquierda. En este sentido, la comparación actúa como lo hace con números de versión, donde se considera que los dígitos más a la izquierda tienen más significado. Por lo tanto, si tiene control sobre el orden de la tupla, debe colocar los elementos más "importantes" antes en la lista.

El tamaño de las tuplas debe ser igual para el resultado que se definirá. Por ejemplo,

```
Q: (1, 2, 3) < (4, 5)
E: The operator "less than" is not defined.
```

Asimismo, los tipos de los elementos de tupla deben coincidir con:

```
Q: (1, 2, 3) < (4, 5, "a")
E: The operator "less than" is not defined.
```

Cualquier comparación con <nothing> causará un error:

```
Q: (nothing) < (nothing)
E: A singular expression is required.
```

```
Q: (nothing) < (1)
E: A singular expression is required.
```

```
Q: (nothing, nothing) < (1)
E: A singular expression is required.
```

```
Q: (nothing, nothing) < (1,2)
E: A singular expression is required.
```

```
Q: (nothing , 1) < (nothing , 2)
```

E: A singular expression is required.

Q: (nothing , 1) < (1,2)

E: A singular expression is required.

## Aritmética

El lenguaje de relevancia incluye las funciones matemáticas binarias típicas, incluidas la suma, la resta, la multiplicación, la división y el módulo.

Q: 21 mod 5

A: 1

Devuelve el entero correspondiente a 21 módulo 5.

Q: 36\*month/2

A: 1 years, 6 months

Multiplica y divide meses que dan como resultado en un tipo 'mes y año'.

Q: 2+3

A: 5

Suma enteros conjuntamente para producir una suma.

Q: current month + 2\*month

A: November

Suma dos meses al mes actual (en este caso, septiembre).

Q: december - current month

A: 3 months

Resta el mes actual (1-12) desde diciembre (12) para producir un tipo 'número de meses'.

Algunos operadores del lenguaje son unarios (requieren un solo argumento), como una negación:

Q: -(3\*5)

A: -15

Como estaba previsto, este signo menos hace desaparecer su argumento (el producto entre paréntesis).

Hay otro símbolo "aritmético", el ampersand (&). Es el operador de concatenación que une cadenas de caracteres:

Q: "now" & "then"

A: nowthen

También se utiliza para crear rangos de tiempo:

Q: now & day

A: Sat, 21 Oct 2006 21:55:28 -0400 to Sun, 22 Oct 2006 21:55:28 -0400

## AND y OR

Los AND y OR lógicos también están disponibles como operadores binarios.

Q: version of regapp "wordpad.exe" as string = "5.1.2600.2180" and name of operating system = "WinXP"

A: True

Devuelve TRUE solo si ambas ecuaciones son verdaderas (expresión AND).

Q: name of operating system = "WinNT" or name of operating system = "WinXP"  
A: True

Devuelve TRUE si un OR de la otra ecuación es verdadero. También puede negar lógicamente una expresión booleana con la palabra clave 'not'.

Q: not exists drive "z:"  
A: True

Devuelve True si la unidad z: no existe. Esta es una operación unaria (que no) se utiliza para negar otro operador unario (que existe).

La utilización de la existencia con la lógica booleana incluso le permite comprobar cosas que de otra forma podrían devolver un error:

Q: Exists folder "C:\doesn't exist" AND Exists files "this should normally break" of folder "c:\doesn't exist"  
A: False

## If-then-else

Las cláusulas if-then-else tienen el formato:

```
if <conditional-expression> then <expression1> else <expression2>
```

Si las sentencias requieren una cláusula 'then' y 'else' o generarán un error.

<expression1> y <expression2> deben tener el mismo tipo, y <conditional-expression> debe ser un booleano singular.

Si <conditional-expression> es true, <expression1> se evaluará y se devolverá; de lo contrario, <expression2> se evaluará y se devolverá.

A partir de la versión 5.1 de IBM Endpoint Manager, se han implementado las cláusulas if-then-else como enlazado a posteriori, por lo que los errores de vocabulario potenciales en la rama que no se tomen se omitirán. Esto asegura la grabación de las expresiones de relevancia entre plataformas sin tener que preocuparse por arrojar errores para inspectores específicos del sistema operativo incorrecto. Por ejemplo, puede escribir:

Q: if name of operating system contains "Win" then name of application "conf.exe" of registry else "conf.exe"  
A: conf.exe  
I: singular string

En un sistema operativo que no sea Windows, esta expresión ejecutará la expresión 'else' y evitará un intento de inspeccionar un registro no existente.

**Nota:** Antes de la versión 5.1 de IBM Endpoint Manager, se han comprobado ambas ramas para asegurarse de que sean significativas, lo que podría generar un error. En ese caso, se podría haber producido un error de análisis en cualquier sistema no Windows cuando se ha detectado la palabra clave desconocida 'registro'.

Las sentencias if-then pueden ser útiles para la creación de errores definidos por el usuario:

Q: if (year of current date as integer < 2006) then "Still good" else error "Expired"  
E: User-defined error: Expired

Esta expresión arroja un error definido por el usuario si el argumento es false.  
 Q: if (name of operating system = "WinXP") then "wired" else if (name of operating system = "WinNT") then "tired" else "expired"  
 A: wired

Esta expresión realiza una prueba de tres maneras del sistema operativo.

## Expresiones

Colocar todas las piezas juntas crea expresiones de relevancia completas. Pueden ser cortas y al grano:

Q: number of active devices  
 A: 156

o pueden ser extremadamente específicas:

Q: exists key whose (value "DisplayVersion" of it as string as version >= "10.0.6626.0" as version AND (character 1 of it = "9" AND (character 2 of it = "0" OR character 2 of it = "1") AND (first 2 of following text of first 3 of it = "11" OR first 2 of following text of first 3 of it = "12" OR first 2 of following text of first 3 of it = "13" OR first 2 of following text of first 3 of it = "28")) AND (preceding text of first "}" of it ends with "6000-11D3-8CFE-0050048383C9")) of name of it) of key "HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall" of registry

Las expresiones de relevancia le permiten analizar e informar de propiedades específicas de los sistemas del cliente con un mínimo de alteraciones, de forma que pueda actuar únicamente en los sistemas que necesiten ayuda y en ninguno de los anteriores.

---

## Inspectores de IBM Endpoint Manager

El lenguaje de relevancia se ha dedicado a manipular los objetos del inspector, que se pueden tener en cuenta como extensiones modulares del lenguaje. Los inspectores están diseñados para interrogar el software, el firmware y el hardware de cada uno de los equipos clientes de la red. Hay miles de inspectores que puede utilizar para analizar diversos aspectos del equipo para asegurarse de que las acciones que propone son destinos adecuados. Los inspectores también se utilizan para generar variables sustituidas en los scripts de acción. Hay bibliotecas del inspector específicas del sistema operativo para Windows, HP-Unix, AIX, Linux, Solaris y Macintosh. Para obtener más información, consulte la *Biblioteca del inspector* para el sistema operativo en el que esté interesado.

Muchas de las palabras clave del lenguaje no son exclusivas; obtienen su significado del contexto. Un contexto del inspector se dicta mediante el formato del inspector. Hay siete formatos:

Formulario	Sintaxis necesaria
Cast	<object> as keyword
Global	palabra clave
Denominado	palabra clave "name" de <object>
NamedGlobal	palabra clave "name"
Numerado	palabra clave <i>number</i> de <object>
NumberedGlobal	palabra clave <i>number</i>

Formulario	Sintaxis necesaria
Sin formato	palabra clave de <object>

Difieren entre sí en formato y sintaxis. Excepto Cast, estos formatos se pueden utilizar para acceder a los objetos simples y a las listas de objetos utilizando el formato plural de la palabra clave.

## Inspectores básicos

Algunos inspectores básicos sirven para expandir el idioma básico. Son similares a elementos de lenguaje de relevancia, pero a menudo dependen del sistema operativo y, por lo tanto, son más fáciles de compilar en las bibliotecas del inspector.

```
Q: floating point "3.14159"
A: 3.14159
I: floating point
```

Crea un número de coma flotante de la cadena de caracteres especificada.

```
Q: string "hello"
A: hello
I: string
```

Crea un tipo de cadena de caracteres de la cadena de caracteres entre comillas especificada.

```
Q: nan of (floating point "1.e-99999" / 0)
A: True
```

Nan (Not A Number) se utiliza para probar los números de coma flotante.

```
Q: first 6 of "Now is the time"
A: Now is
```

Devuelve los primeros caracteres N de la cadena de caracteres especificada.

```
Q: multiplicities of unique values of (1;2;3;3)
A: 1
A: 1
A: 2
```

El inspector de multiplicidad le permite analizar las frecuencias de elementos en una lista.

## Otros ejemplos del inspector

Otros ejemplos de los inspectores básicos incluyen los siguientes:

```
Q: now
A: Thu, 21 Sep 2006 19:39:33 -0400
I: time
```

El inspector 'now' devuelve el día, la fecha, la hora y el huso horario actual desde el sistema cliente. Se trata de un inspector del 'world' (el entorno del sistema del cliente local) que devuelve una hora.

```
Q: exists file "c:/report.txt"
A: True
I: boolean
```

Esta expresión de relevancia devuelve True si existe el archivo especificado. Es un inspector filesystem que evalúa en booleano utilizando el operador de existencia.

```
Q: names of folders of folder "c:/program files"  
A: Adobe  
A: BigFix Enterprise  
A: Cisco Systems  
A: iPod  
A: Macromedia  
A: Microsoft Office ...  
I: string
```

La expresión anterior devuelve una lista de carpetas que residen en la carpeta especificada. Los nombres de las carpetas son tipos de cadenas de caracteres.

```
Q: name of current user  
A: John  
I: string
```

Esta frase devuelve el nombre del usuario actual como un tipo de cadena de caracteres.

## Propagación de no existencia

Si una propiedad no existe, tampoco existirá cualquier otro derivativo de dicha propiedad:

```
Q: exists folder "z:/foo"  
A: False  
Q: files of folder "z:/foo"  
E: Singular expression refers to nonexistent object.  
Q: line 1 of files of folder "z:/foo"  
E: Singular expression refers to nonexistent object.
```

Como la carpeta original no existe, cualquier referencia a la carpeta tampoco existirá.

## Determinación de las propiedades del objeto

El lenguaje de relevancia tiene algunas características incorporadas en él que le ayudan a determinar las propiedades del objeto que puede consultar. Por ejemplo, supongamos que se evalúa la siguiente expresión de relevancia:

```
Q: folder "c:\"  
E: The operator "string" is not defined.
```

Este mensaje de error significa que la expresión de relevancia se ha evaluado satisfactoriamente, pero el lenguaje de relevancia no sabe cómo visualizar una carpeta. Para obtener alguna información de la carpeta, se tienen que consultar sus propiedades. Para ello, se pueden utilizar los 'Introspectores' del lenguaje de relevancia. Los introspectores devuelven información sobre los inspectores que utiliza actualmente el depurador de relevancia y el QnA. Contienen toda la información sobre qué propiedades de un objeto se pueden consultar. Básicamente, son Inspectores del inspector. Por ejemplo, para averiguar las propiedades de una carpeta, se utiliza la consulta:

```
Q: properties of type "folder"  
A: descendants of <folder>: file  
A: file <string> of <folder>: file  
A: folder <string> of <folder>: folder  
A: application <string> of <folder>: application  
A: files of <folder>: file
```

A: find files <string> of <folder>: file  
A: folders of <folder>: folder  
A: security descriptor of <folder>: security descriptor

Sin embargo, no se trata de una lista completa de las propiedades de la carpeta. Un tipo de carpeta también tiene un tipo de padre, el tipo de objeto del sistema de archivos. También se pueden consultar todas las propiedades de un objeto del sistema de archivos. Por ejemplo, nombre de vía de acceso es una propiedad de un objeto del sistema de archivos, pero no se mostraba en la consulta de propiedades anterior. Sin embargo, como la carpeta es un subtipo de un objeto del sistema de archivos, se puede consultar el nombre de la vía de acceso de una carpeta:

Q: pathname of folder "c:\"  
A: c:

Para averiguar si el tipo de carpeta tiene un tipo de padre, utilice la siguiente consulta de relevancia:

Q: parent of type "folder"  
A: filesystem object

La mayoría de los tipos no tienen un tipo de padre. Por ejemplo, los tipos de objeto del sistema de archivos no tienen un tipo de padre.

Q: parent of type "filesystem object"  
A: Singular expression refers to nonexistent object.

Por lo tanto, todas las propiedades que se pueden consultar de una carpeta son propiedades de carpeta o de sistema de archivos, y por lo tanto la siguiente expresión de relevancia listará ambas:

Q: properties of type "folder"; properties of type "filesystem object"  
A: descendants of <folder>: file  
A: file <string> of <folder>: file  
A: folder <string> of <folder>: folder  
...  
A: normal of <filesystem object>: boolean  
A: temporary of <filesystem object>: boolean  
A: compressed of <filesystem object>: boolean  
A: offline of <filesystem object>: boolean  
...

Se puede descubrir una lista de propiedades aún más completa mediante la siguiente expresión:

Q: properties whose ( it as string contains "folder" )  
A: ancestors of <filesystem object>: folder  
A: descendants of <folder>: file  
A: parent folder of <filesystem object>: folder  
...  
A: application folder <string> of <registry>: folder  
A: application folder of <registry key>: folder  
A: application folder <string> of <registry key>: folder  
A: install folder <integer>: folder

---

## Relevancia en el análisis de propiedades

### Visualización de análisis de propiedades

Desde la consola de IBM Endpoint Manager, pulse en un elemento desde el separador **Análisis**. En la ventana de la parte inferior, pulse en el separador **Detalles**. Aquí puede ver las expresiones de relevancia detrás de un análisis de propiedad.

Por ejemplo, seleccione **Versiones del componente de BES** desde la lista Análisis. Pulse en el separador Detalles para ver las expresiones de relevancia detrás de cada análisis. Estos Análisis devuelven el cliente, el retransmisor, la consola y la versión del servidor de cada equipo cliente. Por ejemplo, la Versión del retransmisor de BES tiene la siguiente declaración de relevancia:

```
if (exists regapp "BESRelay.exe") then version of regapp "BESRelay.exe" as string  
else "Not Installed"
```

Esto devuelve la versión del retransmisor de BES, tras determinar que exista. Si no es así, devuelve 'Not Installed'.

Una propiedad puede devolver más de un solo elemento. Puede, por ejemplo, devolver una tupla:

```
(total run count of it, first start time of it, last time seen of it, total duration  
of it) of application usage summaries "excel.exe"
```

Esta cláusula de relevancia devuelve varias propiedades que resumen el uso del cliente de Excel.

## Creación de análisis de propiedades

Puede crear sus propias propiedades. Estas le permiten realizar un seguimiento de cualquier combinación de software, hardware y firmware que desee, en toda la red. Una vez creados, los resultados del análisis de la propiedad se pueden imprimir o representar gráficamente.

Por ejemplo, es posible que desee supervisar el estado de los lenguajes del sistema operativo en toda la red. Podría utilizar una cláusula de relevancia como esta para recuperar la información:

```
Q: system language  
A: English (United States)  
I: singular string
```

Puede otorgar a esta expresión un nombre del que puede realizar un seguimiento, como 'Lenguaje del sistema'. Fíjese cómo:

Desde el menú **Herramientas**, seleccione **Crear nuevo análisis**. Escriba el título y la descripción del análisis y, a continuación, pulse en el separador **Propiedades**. Escriba su nombre deseado, la expresión de relevancia y el periodo de evaluación en los campos adecuados. Una vez activado, se informará de este análisis al servidor de IBM Endpoint Manager, lo que le permite ver o trazar un gráfico de los resultados.

---

## Relevancia en los scripts de la acción

### Visualización de scripts de acción

Puede ver un script de acción de la consola de IBM Endpoint Manager seleccionando un Fixlet o una tarea y pulsando en el separador **Detalles**. Para obtener más información en la sintaxis de acción, consulte la *Referencia del lenguaje de acciones de IBM Endpoint Manager*.

En muchos de los scripts de acción, puede ver las expresiones de relevancia dentro de llaves {}. Cuando se ejecuta la acción, se evaluarán estas expresiones en cada

sistema cliente y los resultados se sustituirán en el script de la acción. Esto permite a un autor crear acciones que están personalizadas para cada cliente. Por ejemplo:

```
run "{pathname of regapp "excel.exe"}"
```

Este ejemplo puede ejecutar un programa sin saber dónde se encuentra. La expresión de relevancia entre corchetes evalúa el nombre de vía de acceso automáticamente utilizando el inspector regapp. La incorporación de expresiones de relevancia le permite ejecutar de forma precisa scripts de acción de destino. Este script puede utilizar un nombre de vía de acceso diferente en cada cliente, pero continuará funcionando según lo esperado. Esto le permite grabar scripts legibles y compactos que personalizarán automáticamente ellos mismos a cada cliente de la red.

Además de sustituir variables, puede utilizar expresiones de relevancia para realizar aserciones que puedan modificar el flujo del código:

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

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

La sustitución no es recursiva, aunque cualquier mandato concreto puede tener una o varias expresiones para evaluar antes de la ejecución. La aplicación IBM Endpoint Manager espera encontrar una expresión única dentro de las llaves. Si ve otra llave izquierda antes de que detecte una llave derecha de cierre, la trata como un carácter normal:

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

podría enviar esta cadena de caracteres para salida:

```
a left brace: {
```

Por lo tanto, no son necesarios caracteres de escape especiales para representar una llave a la izquierda. Para extraer una llave derecha literal sin finalizar la sustitución, utilice un carácter doble:

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

podría enviar esta cadena de caracteres para salida:

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

O piense en este ejemplo:

```
appendfile {{ name of operating system } {name of operating system}}
```

Cuando se analice este ejemplo, las llaves dobles a la izquierda indican que lo que sigue *no* es una expresión de relevancia. Por lo tanto, esta parte del script se trata como una cadena de caracteres, hasta la primera llave derecha. La primera llave izquierda indica el inicio real de una expresión de relevancia. Se genera la siguiente línea en `__appendfile`:

```
{ name of operating system } WinXP
```

## Creación de scripts de acción

Puede crear sus propios scripts de acción en la consola de IBM Endpoint Manager seleccionando **Llevar a cabo una acción personalizada** desde el menú **Herramientas**. Se abrirá un cuadro de diálogo donde puede establecer la Relevancia, el mensaje, etc. Pulse en el separador **Script de acción**. En la ventana

de texto que aparece, puede especificar cualquier script de acción que desee. Puede incorporar los resultados de una expresión de relevancia en cualquier lugar del script indicándola entre llaves {}.

Por ejemplo, es posible que desee descargar un archivo con un mandato como:  
download http://download.bigfix.com/download/bes/60/BESServerUpgrade-6.0.12.5.exe

A continuación, para asegurarse de que lo descarga correctamente con un hash seguro, puede añadir este mandato con una cláusula de relevancia incorporada:

```
continue if {(size of it = 18455939 AND sha1 of it =  
"58a879f5b98357c4ec6f5ff7dbe3307eef5ca2ec") of file "BESServerUpgrade-6.0.12.5.exe"  
of folder "__Download"}
```

Esta expresión compara la longitud del archivo (que se encuentra consultando en la carpeta \_\_Download) con un tamaño conocido. También compara el sha1 del archivo con un valor conocido. Esta construcción le permite detener la ejecución del script de acción si el archivo no se ha descargado correctamente. Ilustra un uso común del lenguaje de relevancia para realizar una aserción en un script de acción.

Es posible que desee definir una clave de registro con los resultados de una expresión de relevancia:

```
regset "{[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess  
\Parameters\FirewallPolicy\" & (if (current profile type of firewall = domain  
firewall profile type) then ("DomainProfile") else ("StandardProfile")) & "]" }"  
"EnableFirewall"=dword:00000000
```

Aquí, el resultado de la expresión de relevancia en las llaves se sustituye en el nombre del valor de registro. Este ejemplo muestra cómo se ramifica en función del valor de un inspector a fin de establecer un registro con la cadena de caracteres adecuada.

También puede establecer variables de acción con los resultados de las declaraciones de relevancia. Se lleva a cabo con el mandato de parámetro:

```
parameter "tmpfolder" = "{pathname of folder (value of variable "tmp" of  
environment)}"
```

Debido a que la expresión de relevancia utilizada para marcar como objetivo al Fixlet es a menudo la misma que la de la acción correspondiente, es más que probable que esté resolviendo el problema correcto. Esto hace que la escritura de scripts sea más fácil y hace a los scripts más sólidos y precisos.

---

## Expresiones de relevancia útiles

Esta sección contiene algunas cláusulas de relevancia reales que son fáciles de personalizar para su propio uso. En estos ejemplos, puede ver cómo puede crear una expresión compleja de algunos elementos básicos y simples.

## Utilización de rangos de números

A partir de la versión 8 de TEM, puede utilizar rangos de números de varias maneras. Para generar los primeros cuatro números, puede utilizar una expresión como la siguiente:

Q: integers to (3)  
A: 0  
A: 1  
A: 2  
A: 3

Tenga en cuenta que el índice se basa en cero, por lo que el rango va de 0 a 3, no de 1 a 3. También puede utilizar números negativos:

Q: integers in (-2, 2)  
A: -2  
A: -1  
A: 0  
A: 1  
A: 2

Además de listar números incrementalmente de uno en uno, también puede especificar un intervalo de paso:

Q: integers in (-10, 3, 3)  
A: -10  
A: -7  
A: -4  
A: -1  
A: 2

Esta expresión pasa del primer valor (-10) en intervalos de tres, deteniéndose en el primer entero que no supera el valor final. Puede derivar el valor final de la longitud de una cadena de caracteres, como en este ejemplo:

Q: integers in ((0, length of it, 2) of "00000b000100")  
A: 0  
A: 2  
A: 4  
A: 6  
A: 8  
A: 10  
A: 12

Tenga en cuenta que la longitud es 12, y que el mandato integers incluye los puntos finales, por lo que incluye un puntero al final de la cadena de caracteres.

Q: concatenation "," of substrings (integers in (0, length of it, 2), 2) of "00000b000100020000000000000000"  
A: 00,00,0b,00,01,00,02,00,00,00,00,00,00,00,00

A continuación se muestra un ejemplo más complicado que combina dos listas. La primera lista es numérica (1, 2, 3) y la segunda lista es alfabética (a, b, c). Esta expresión combina las dos para crear (1a, 2b, 3c):

Q: substrings (integers in (0, length of it, 2) , 2) of (concatenation of substrings ((item 0 of it ; item 0 of it + item 1 of it) of (positions of first ((length of it - 1) / 2) of it, length of it / 2), 1) of it) of concatenation of ("1";"2";"3"); ("a";"b";"c"))  
A: 1a  
A: 2b  
A: 3c

La expresión realiza un uso correcto de la sintaxis "item N of it" para indexar en una lista de objetos.

## Utilización del operador de barra

Desde la versión 8 de TEM, puede utilizar el operador de barra ( | ) para iterar a través de una lista y buscar el primer elemento que no producirá un error. En el ejemplo siguiente, se da por supuesto que el archivo gone.txt no existe, por lo que obtener su tamaño emite un error:

```
Q: size of file "c:\gone.txt" | 10
A: 10
```

La expresión omite el error del archivo y llega a 10.

```
Q: (size of file "c:\gone.txt" > 0) | false
A: False
```

El operador de barra busca errores, no true o false, de modo que informará de false si es la primera frase que no tiene errores.

```
Q: ( (not exists file "c:\gone.txt") | (size of file "c:\gone.txt" = 0) )
A: False
```

Como el operador de barra puede pasarse por alto a través de errores, puede simplificar el código. Normalmente se comprueba la existencia de cada archivo antes de examinarlo:

```
(
  if
  (
    exists result (it,bes properties "Shared Groups")
    whose (exists value of it)
  )
  then
    value of result (it,bes properties "Shared Groups")
  else
    "No Result"
)
of bes computers
```

Este es el método por el que puede grabarlo utilizando el operador de barra:

```
(
  value of result (it,bes property "Shared Groups") | "No Result"
)
of bes computers
```

El operador de barra sólo mantiene la iteración a través de la lista hasta que alcanza una frase sin errores, que en este caso es el mensaje "No Result".

## Manipulación de cadenas

### Creación de varios resultados

Se pueden extraer varias subcadenas de caracteres de una cadena de caracteres con comas (,) como delimitadores:

```
Q: substrings separated by ", " of "apple, orange, pear, kiwi"
A: apple
A: orange
A: pear
A: kiwi
I: plural substring
```

O si lo desea, utilice únicamente espacios:

```
Q: substrings separated by " " of "apple orange pear kiwi"
```

## Inversión de una cadena de caracteres

Si conoce la longitud de la cadena de caracteres, puede invertir explícitamente el orden de los caracteres:

Q: concatenation of characters (4; 3; 2; 1; 0) of "abcde"  
A: edcba

Utiliza las posiciones de los caracteres en orden inverso para invertir la cadena de caracteres. ¿Pero qué ocurre si no sabe cuántos caracteres hay en la cadena de caracteres? Hay algunas propiedades de las cadenas de caracteres que se pueden activar en la tarea:

Q: positions of "abcde"  
A: 0  
A: 1  
A: 2  
A: 3  
A: 4  
A: 5

Indica que hay seis posiciones en la cadena de caracteres (incluyendo el puntero al final de la cadena de caracteres), que se corresponde con el número de caracteres, más uno. A medida que explora la cadena de caracteres, hay cada vez menos caracteres tras la posición especificada:

Q: following texts of (positions of "abcde")  
A: abcde  
A: bcde  
A: cde  
A: de  
A: e  
A:

La longitud de estas cadenas de caracteres puede pedirse, para producir una lista de números que es una inversión perfecta de las posiciones listadas anteriormente.

Q: lengths of (following texts of (positions of "abcde"))  
A: 5  
A: 4  
A: 3  
A: 2  
A: 1  
A: 0

Esta lista invertida se puede utilizar para explorar en orden inverso a través de la cadena de caracteres, concatenando a medida que avanza:

Q: concatenation of characters (lengths of (following texts of (positions of it))) of "abcde"  
A: edcba

## Índice de subcadena de cadena de caracteres

Desde la versión 8 de TEM, puede recuperar los valores de subcadena de caracteres indexados:

Q: substring(0, 3) of "abcdefgh"  
A: abc

Esta expresión devolverá los primeros tres caracteres de la cadena de caracteres especificada. El valor inicial está basado en cero, por lo que para encontrar una subcadena de caracteres que comience en el cuarto carácter, utilice una expresión como esta:

Q: substring(3, 3) of "abcdefgh"  
A: def

Puede concatenar las subcadenas de caracteres que encuentre también. Por ejemplo, para producir una serie de bytes hexadecimales desde una cadena de caracteres hexadecimales, utilice una expresión como la siguiente:

Q: concatenation "," of substrings (integers in (0, length of it, 2), 2) of "00000b000100020000000000000000"  
A: 00,00,0b,00,01,00,02,00,00,00,00,00,00,00,00,00

Esta expresión utiliza el mandato **integers in**, que devuelve una lista de números pares con un número de inicio (a partir de 0) y un intervalo de paso (2):

Q: (integers in (0, length of it, 2), 2) of "123456"  
A: 0, 2  
A: 2, 2  
A: 4, 2  
A: 6, 2

Como aquí deseamos valores hexadecimales de dos dígitos, añadimos el ",2" a los valores de inicio.

## Manipulación de fechas y horas

### Conversión del tiempo a mm/dd/aaaa

Para convertir el formato de tiempo devuelto por el inspector "now" al formato mm/dd/aaaa, extraiga los tres componentes (mes, día, año) y, a continuación, concátelos con barras inclinadas. Comience con la parte *date* de now (excluye la parte time):

Q: date (local time zone) of now  
A: Mon, 25 Sep 2006  
I: date

Ello devuelve una fecha con los elementos que deseamos reorganizar. El inspector "month" puede proporcionarnos un mes numérico correctamente formateado (dos dígitos, con un cero inicial):

Q: (month of date (local time zone) of now) as two digits  
A: 09  
I: string

El inspector "day\_of\_month" devuelve la fecha, que formateamos como se mostró anteriormente:

Q: day\_of\_month of date (local time zone) of now as two digits  
A: 25  
I: day of month

El inspector year redondea todo:

Q: year of date (local time zone) of now as string  
A: 2006  
I: string

Concatene estos componentes con barras inclinadas para finalizarlo:

Q: (month of date (local time zone) of now) as two digits & "/" & day\_of\_month of date (local time zone) of now as two digits & "/" & year of date (local time zone) of now as string  
A: 09/25/2006  
T: 0.263 ms  
I: string

Se puede mejorar considerablemente invocando la función "now" sólo una vez y haciendo referencia a ella en otro lugar con la palabra clave "it":

```
Q: (month of it as two digits & "/" & day_of_month of it as two digits
& "/" & year of it as string) of date (local time zone) of now
A: 09/25/2006
T: 0.170 ms
I: string
```

Esta versión es más corta, más fácil de leer y cerca de una tercera parte más rápida. Quizá de más importancia sea que el valor de now puede cambiar entre invocaciones, por lo que puede obtener una respuesta incorrecta con la primera técnica. En Nochevieja, por ejemplo, es posible que obtenga diciembre conectado con el año incorrecto.

## Conversión de aaaammdd a la fecha

La conversión de aaaammdd a un formato de fecha estándar utiliza un conjunto distinto de inspectores. En primer lugar, rompa la cadena de caracteres en las partes día, mes y año:

```
Q: first 2 of following text of position 6 of "20071201" as integer
A: 1
Q: first 2 of following text of position 4 of "20071201" as integer
A: 12
Q: first 4 of "20071201" as integer
A: 2007
```

A continuación, convierta estos enteros en sus tipos de fecha de componentes:

```
Q: day_of_month (first 2 of following text of position 6 of "20071201")
A: 1
I: day of month
Q: month (first 2 of following text of position 4 of "20071201" as integer)
A: December
I: month
Q: year (first 4 of "20071201" as integer)
A: 2007
I: year
```

Estos componentes se concatenarán para crear una fecha estándar:

```
Q: day_of_month (first 2 of following text of position 6 of "20071201" as integer)
& month (first 2 of following text of position 4 of "20071201" as integer) & year
(first 4 of "20071201" as integer)
A: Sat, 01 Dec 2007
I: date
```

Se puede simplificar utilizando la palabra clave 'it' como una variable que representa "20071201":

```
Q: (day_of_month (first 2 of following text of position 6 of it
as integer) & month (first 2 of following text of position 4 of it
as integer) & year (first 4 of it as integer)) of "20071201"
A: Sat, 01 Dec 2007
I: date
```

Se puede conseguir un resultado similar utilizando una expresión regular. La fecha se puede extraer seleccionando los primeros cuatro dígitos:

```
Q: parenthesized part 1 of ( matches (regex "(\\d\\d\\d\\d)(\\d\\d)(\\d\\d)" )
of "20051201")
A: 2005
```

Los diversos segmentos de fecha se pueden montar junto a estas líneas para crear:

Q: (day\_of\_month (parenthesized part 3 of it as integer) & month (parenthesized part 2 of it as integer) & year (parenthesized part 1 of it as integer))of (matches (regex "(\\d\\d\\d\\d)(\\d\\d)(\\d\\d)") of "20051201")  
A: Thu, 01 Dec 2005

Igual que antes, esta expresión utiliza el inspector 'day\_of\_month' para devolver una fecha correspondiente a la concatenación de los componentes.

## Suma de unidades de tiempo

Desde la versión 8 de TEM, puede sumar distintas unidades de tiempo para crear un valor de tiempo complejo. Esto le permite trabajar con una mayor granularidad de tiempo y una mayor precisión. A continuación, se muestran algunos ejemplos:

Q: sum of (second; minute; hour; day)  
A: 1 day, 01:01:01  
Q: sum of ((second \* 5); (minute \* 3))  
A: 00:03:05  
Q: sum of ((second \* 1); (second \* 2); (minute \* 3); (minute \* 4); (hour \* 5); (hour \* 6); (day \* 7); (day \* 8))  
A: 15 days, 11:07:03  
Q: sum of ((second \* 5); (minute \* 3); (hour \* 17); (day \* -23))  
A: -22 days, 06:56:55  
Q: sum of (second / 10)  
A: 00:00:00.100  
Q: sum of ((hour + hour))  
A: 02:00:00

Tenga en cuenta que la suma de un intervalo de tiempo sólo incluye semanas, días, horas, minutos y segundos. No sumará meses ni años.

## Operación en listas de números de coma flotante

Desde la versión 8 de TEM, puede sumar varios números de coma flotante en una lista separada por punto y coma:

Q: sum of ("1.2" as floating point; "1.5" as floating point)  
A: 2.7

Es como añadir números de coma flotante, pero permite sumar una lista de varios números:

Q: sum of ("1.0" as floating point; "1.1" as floating point; "1.9" as floating point)  
A: 4.0

No es que, como con la suma de coma flotante normal, deba prestar atención a los dígitos de precisión que utiliza. El número menos preciso de la suma determina la precisión de la respuesta. Si sólo necesita un dígito a la derecha de la coma decimal, puede indicarlo con una frase como esta:

Q: sum of ("1.1" as floating point; "1.06" as floating point)  
A: 2.2

Si necesita más dígitos de precisión, añada ceros.

Q: sum of ("1.10" as floating point; "1.06" as floating point)  
A: 2.16

Puede aumentar explícitamente la precisión con el operador **more significance**:

Q: sum of (more significance 1 of floating point "1.1"; "1.06" as floating point)  
A: 2.16

También puede multiplicar una lista de números con el operador product:  
Q: product of ("1" as floating point; "2" as floating point; "3" as floating point)  
A: 6

## Utilización de búsquedas de comodín para archivos y carpetas

Desde la versión 8 de TEM, puede utilizar los inspectores find para buscar archivos y carpetas con búsquedas de comodín:

Q: names of find folders "system\*" of windows folder  
A: system  
A: system32

Esta expresión devuelve todas las carpetas de la carpeta Windows que tengan "system" en su nombre. También puede sumar la búsqueda para encontrar el número de carpetas relevantes:

Q: number of names of find folders "\*86\*" of folder "C:\windows\winsxs"  
A: 5050

Esta expresión ha encontrado 23 archivos con "86" en el nombre. Aquí, los asteriscos son caracteres comodín que sobresalen para cualquier conjunto de caracteres (incluido nulo), y encontrará archivos denominados "86skidoo", "my86th" y "last86".

También puede utilizar el comodín "?", que coincide con un carácter individual y, por lo tanto, es más restrictivo:

Q: number of names of find folders "?86\*" of folder "C:\windows\winsxs"  
A: 4325

Puede utilizar varios caracteres para expandir la búsqueda:

Q: number of find folders "win\*" of find folders "system\*" of windows folder  
A: 3

Puede buscar la existencia de cualquier nombre de archivo utilizando un solo asterisco:

Q: number of names of find files "\*" of find folders "X86\*" of find folders "win\*" of windows folder  
A: 51

Esta expresión busca todos los archivos de la unidad c: raíz.

Q: find files "\*" of folder "c:"  
A: "AUTOEXEC.BAT" "" "" "" ""  
A: "boot.ini" "" "" "" ""  
A: "CONFIG.SYS" "" "" "" ""  
A: "IO.SYS" "" "" "" ""  
A: "MSDOS.SYS" "" "" "" ""  
A: "NTDETECT.COM" "" "" "" ""  
A: "ntldr" "" "" "" ""  
A: "pagefile.sys" "" "" "" ""  
A: "test.txt" "" "" "" ""

## Comparación de versiones

Las comparaciones de versiones numéricas pueden ser delicadas, porque no son números en el sentido tradicional. Los números de versión normalmente tienen varios segmentos separados por puntos, como "6.01.2.3". Una estructura común (pero no universal) numera los releases de la siguiente manera:

major.minor[.revision[.build]]

Por lo tanto, cuando se comparan versiones, debe especificar todos los segmentos relevantes para obtener una comparación adecuada. Si las compara como si fueran enteros o números de coma flotante, es posible que obtenga la respuesta incorrecta. Tenga en cuenta estos ejemplos:

Q: "6" as version < "6.44" as version  
A: False  
Q: "6.0" as version < "6.44" as version  
A: True

Funciona la segunda expresión de relevancia, porque tiene el mismo número de segmentos de versión, de modo que se compara adecuadamente.

Q: "5" as version = "5.50" as version  
A: True  
Q: "5.00" as version = "5.50" as version  
A: False

La segunda expresión falla correctamente, porque compara una versión de dos segmentos con otra versión de dos segmentos.

No dé por supuesto que los segmentos de la versión son de dos dígitos:

Q: "5.100" as version > "5.99" as version  
A: True  
Q: "5.10" as version > "5.99" as version  
A: False

El lenguaje de relevancia compara los valores numéricos de los segmentos de la versión (separados por puntos), independientemente del número de dígitos del segmento. Para estar seguro, especifique siempre los números de versión completa.

## Inspección del registro de Windows

Puede abreviar las claves raíces del registro. Por ejemplo, en lugar de "HKEY\_LOCAL\_MACHINE\Software", puede escribir "HKLM\Software", etc. A continuación se muestra la lista completa de los accesos directos del registro:

HKCR	HKEY_CLASSES_ROOT
HKCU	HKEY_CURRENT_USER
HKLM	HKEY_LOCAL_MACHINE
HKU	HKEY_USERS
HKCC	HKEY_CURRENT_CONFIG

### Acceso a las claves de usuarios actuales

El cliente de IBM Endpoint Manager se ejecuta como la cuenta de LOCAL SYSTEM y, por lo tanto, su rama HKEY\_CURRENT\_USER no coincide con lo iniciado en la rama del usuario. Sin embargo, todavía es posible obtener lo iniciado en la rama HKEY\_CURRENT\_USER del usuario de HKEY\_USERS buscando a través de las claves de inicio de sesión el nombre del usuario actual:

Q: name of key whose ((it = name of current user as lowercase OR it starts with name of current user as lowercase & "0") of (it as string as lowercase) of value "Logon User Name" of key "Software\Microsoft\Windows\CurrentVersion\Explorer" of it) of key "HKEY\_USERS" of registry  
A: S-1-5-21-1214450339-2025729265-839522115-1013

Asegúrese de incluir la palabra "key" cuando desee examinar una clave de registro. Es fácil buscar en una expresión como esta y pensar que está obteniendo una respuesta válida:

```
Q: exists "HKEY_LOCAL_MACHINE\Software\Microsoft\Active Setup\Installed Components
\{f502aef4-a754-4c82-9f12-a5149f71ea89}" of registry
A: True
```

Sin embargo, lo que es verdadero aquí es que existe el literal de la cadena de caracteres, no la clave. La expresión correcta es:

```
Q: exists key "HKEY_LOCAL_MACHINE\Software\Microsoft\Active Setup
\Installed Components\{f502aef4-a754-4c82-9f12-a5149f71ea89}" of registry
A: False
```

## Búsqueda de valores y claves de registro

Este es el procedimiento para iterar a través de los nombres y de los valores de claves del registro:

```
Q: (names of it, it) of values of key "HKLM\SOFTWARE\Microsoft\Windows
\CurrentVersion\Installer\UserData\S-1-5-18\Products\FAA9C0AB723160F34
BC943B9B48E7F04\Patches\30BCFA5611132E741BDE91D8B6B09CE6" of registry
A: MSI3, 1
A: State, 1
A: Uninstallable, 1
A: LUAEnabled, 0
A: PatchType, 1
A: Installed, 20110106
A: DisplayName, KB945140
```

A: MoreInfoURL, <http://www.microsoft.com/>

Puede probar la existencia de claves específicas:

```
Q: exists value whose (name of it is "DisplayName") of key "HKEY_LOCAL_MACHINE
\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products
\FAA9C0AB723160F34BC943B9B48E7F04\Patches\30BCFA5611132E741BDE91D8B6B09CE6"
of registry
A: True
```

Y puede probar valores específicos de una clave:

```
Q: exists value whose (name of it is "DisplayName" and it is "KB945140") of key
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData
\S-1-5-18\Products\FAA9C0AB723160F34BC943B9B48E7F04\Patches\30BCFA5611132E741BDE
91D8B6B09CE6" of registry
A: True
```

## Detección de las unidades de red asignadas

Es fácil y rápido buscar los nombres de las unidades conectadas al sistema local:

```
Q: names of drives
A: A:
A: C:
A: D:
A: E:
A: F:
A: G:
```

Pero, ¿cómo obtener información sobre las unidades asignadas?

```
Q: (selects ("ProviderName from win32_LogicalDisk") of WMI)
A: ProviderName
A: ProviderName
```

```
A: ProviderName
A: ProviderName
A: ProviderName
A: ProviderName=\\Plato\shared docs
```

La utilización de un inspector WMI como el anterior muestra que la última unidad se asigna a una carpeta docs compartida. También puede correlacionar los nombres de unidad a los nombres compartidos:

```
Q: (if property "ProviderName" of it as string contains "=" then (substring
after "=" of (property "Name" of it as string) & " -- " & substring after "="
of (property "ProviderName" of it as string)) else nothing) of select objects
("Name,ProviderName from win32_LogicalDisk")of WMI
A: G: -- \\Plato\shared docs
```

Esta expresión localiza todas las unidades asignadas, y devuelve sus nombres y su asignación.

## Búsqueda del tamaño de archivo de paginación

La siguiente expresión devolverá el tamaño de archivo de paginación en megabytes:

```
Q: sum of (preceding texts of firsts " " of following texts of firsts ".sys " of
preceding texts of firsts "%00" of following texts of substrings "%00" of ("%00"
& value "PagingFiles" of key "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control
\Session Manager\Memory Management" of registry as string) as integer)
A: 1536
```

## Detección de servicios

Lo siguiente devolverá el nombre de visualización e 'imagepath' de los servicios que se deben listar en el control de servicios:

```
Q: (display name of it, image paths of it) of services
A: Alerter, %SystemRoot%\system32\svchost.exe -k LocalService
A: Application Layer Gateway Service, %SystemRoot%\System32\alg.exe
A: Application Management, %SystemRoot%\system32\svchost.exe -k netsvcs
A: ASP.NET State Service, %SystemRoot%\Microsoft.NET\Framework\v1.1.4322
\aspnet_state.exe
A: Windows Audio, %SystemRoot%\System32\svchost.exe -k netsvcs
A: Background Intelligent Transfer Service, %SystemRoot%\system32\svchost.exe
-k netsvcs
A: Computer Browser, %SystemRoot%\system32\svchost.exe -k netsvcs...
```

Consulte la referencia Documentación de registro de Microsoft para obtener más información.

## Búsqueda de la última hora de grabación de las claves de registro

Puede detectar la última vez que se ha grabado cualquier clave de registro determinada:

```
Q: last write time of key "HKEY_LOCAL_MACHINE\SOFTWARE\BigFix\" of registry
A: Thu, 07 Jan 2010 17:32:08 -0800
```

## Variables de entorno

Las variables de entorno de Windows están disponibles para la inspección utilizando una frase como esta:

Q: value of variable "PATH" of environment  
A: /usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:  
/sbin:/bin:/usr/sbin:/usr/bin:/root/bin

Puede encontrar y devolver todas las variables de entorno con una expresión sencilla:

Q: (names of it, values of it) of variables of environment  
A: BESClientactionMastheadPath, /etc/opt/BESClient/actionsite.afxm  
A: BESClientConfigPath, /var/opt/BESClient/besclient.config  
A: CVS\_RSH, ssh  
A: DISPLAY, :0  
A: G\_BROKEN\_FILENAMES, 1  
A: HISTSIZE, 1000  
A: HOME, /root  
A: HOSTNAME, t-rhel5-x86-1  
A: INPUTRC, /etc/inputrc  
A: KDEDIR, /usr  
A: KDE\_IS\_PRELINKED, 1  
A: KDE\_NO\_IPV6, 1  
A: LANG, en\_US.UTF-8  
A: LD\_LIBRARY\_PATH, /opt/BESClient/bin  
A: LESSOPEN, |/usr/bin/lesspipe.sh %25s  
A: LOGNAME, root

A partir de la versión 8 de TEM, también puede inspeccionar las variables de entorno asociadas con procesos en máquinas \*nix:

Q: variables of environment of process 2186  
A: BESClientactionMastheadPath = /etc/opt/BESClient/actionsite.afxm  
A: BESClientConfigPath = /var/opt/BESClient/besclient.config  
A: BESRelayConfigPath =  
A: CONSOLE = /dev/pts/0  
A: DISPLAY = :0

## Determinación de la información del sistema operativo

Hay inspectores que pueden interrogar el sistema operativo. Por ejemplo, para encontrar la versión, puede utilizar una expresión como esta:

Q: version of operating system  
A: 6.1.7600

Para recuperar el idioma del sistema operativo Windows actual, utilice:

Q: language of version block of file "user32.dll" of system folder  
A: English (United States)

Esta cláusula de relevancia se aplica a todas las versiones existentes de Windows, incluida 95, 98, ME, XP, NT4 y 2000.

Para buscar el nombre del sistema operativo del cliente, puede utilizar una sentencia como la siguiente:

Q: name of operating system as lowercase contains "win"  
A: True

A partir de TEM versión 8, hay un atajo para esta sintaxis.

Q: windows of operating system  
A: True

Los valores aceptables son windows, mac y unix.

## Acceso al Administrador de tareas (Procesos)

Puede acceder al administrador de tareas en un sistema Windows para examinar todos los procesos en ejecución. A continuación se muestra un ejemplo que devuelve varias propiedades del proceso específico denominado SQLAGENT:

```
Q: (working set size of it / 1024, page file usage of it / 1024, user of it)
of processes whose (name of it is "SQLAGENT.EXE")
A: 6492, 9336, TESTBOX\Administrator
```

Puede examinar prácticamente cada propiedad de procesos en ejecución:

```
(name of it,
(if exists image file of it then pathname of image file of it else "n/a"),
id of it,
page fault count of it,
peak working set size of it,
working set size of it,
quota peak paged pool usage of it,
quota paged pool usage of it,
quota peak nonpaged pool usage of it,
quota nonpaged pool usage of it,
page file usage of it,
peak page file usage of it,
io read count of it,
io write count of it,
io other count of it,
io read size of it,
io write size of it,
io other size of it,
creation time of it,
kernel time of it,
user time of it,
handle count of it,
session id of it,
wow64 of it,
user of it,
(if (base priority of it = normal priority) then "Normal" else
if (base priority of it = high priority) then "High" else
if (base priority of it = idle priority) then "Idle" else
if (base priority of it = realtime priority) then "Realtime" else
if (base priority of it = above normal priority) then "Above Normal" else
if (base priority of it = below normal priority) then "Below Normal" else "N/A")
)
of processes
```

Al ejecutar esta expresión en el depurador de Fixlet, utilice el separador Cláusula única (**Archivo > Nuevo separador > Nuevo separador de cláusula única**). Iterará a través de los procesos en ejecución, lo que devolverá cadenas como esta:

```
smss.exe, C:\Windows\System32\smss.exe, 284, 540, 1175552, 69632, 38360, 10648,
3568, 1624, 454656, 520192, 12, 4, 654, 53790, 4, 10990, ( Wed, 04 Apr 2012 11:06:35
-0400 ), 00:00:00.031200, 00:00:00.015600, 30, 0, False, NT AUTHORITY\SYSTEM, Normal
csrss.exe, C:\Windows\System32\csrss.exe, 408, 2778, 4558848, 1863680, 155376,
143920, 15368, 13088, 2154496, 2154496, 497, 0, 5590, 386796, 0, 262035, ( Wed,
04 Apr 2012 11:06:40 -0400 ), 00:00:01.404009, 00:00:00.218401, 646, 0, False,
NT AUTHORITY\SYSTEM, Normal
wininit.exe, C:\Windows\System32\wininit.exe, 480, 1351, 4579328, 167936, 106368,
102384, 13960, 9736, 1531904, 1847296, 1, 0, 1848, 7168, 0, 9374, ( Wed, 04 Apr
2012 11:06:45 -0400 ), 00:00:00.093600, 00:00:00.046800, 77, 0, False,
NT AUTHORITY\SYSTEM, High
```

## Examen de tareas planificadas y de ejecución

Se puede acceder y examinar las tareas de ejecución y planificadas en un sistema Windows a partir de la versión 8 de TEM. Por ejemplo, puede buscar los nombres de tareas actualmente en ejecución con una expresión como esta:

```
Q: names of running tasks
A: SystemSoundsService
A: MsCtfMonitor
```

Para ver qué tareas planifican su ejecución y cuándo, utilice una expresión como la siguiente:

```
Q: (names of it, paths of it, next run times of it) of scheduled tasks
A: AD RMS Rights Policy Template Management (Automated), \Microsoft
\Windows\Active Directory Rights Management Services Client\AD RMS Rights
Policy Template Management (Automated), ( Thu, 19 Aug 2010 03:05:32 -0700 )
A: AitAgent, \Microsoft\Windows\Application Experience\AitAgent,
( Thu, 19 Aug 2010 02:30:00 -0700 )
A: ProgramDataUpdater, \Microsoft\Windows\Application Experience
\ProgramDataUpdater, ( Thu, 19 Aug 2010 00:30:00 -0700 )
A: Consolidator, \Microsoft\Windows\Customer Experience Improvement
Program\Consolidator, ( Wed, 18 Aug 2010 22:00:00 -0700 )
A: KernelCeipTask, \Microsoft\Windows\Customer Experience
Improvement Program\KernelCeipTask, ( Thu, 19 Aug 2010 03:30:00 -0700 )
```

El ejemplo siguiente muestra cómo listar todas las tareas que se ejecutarán al inicio de sesión:

```
Q: names of scheduled tasks whose (exists definitions whose (exists
triggers whose (type of it is logon task trigger type) of it) of it)
A: AD RMS Rights Policy Template Management (Automated)
A: AD RMS Rights Policy Template Management (Manual)
A: Microsoft-Windows-DiskDiagnosticResolver
A: Logon Synchronization
A: AutoWake
```

## Reconocimiento de Service Packs de Office

Es posible que tenga que verificar un Service Pack de Office antes de aplicar una acción. No se trata siempre de un procedimiento trivial. La información sobre Office XP se almacena como una clave de desinstalación en el registro, que tiene un nombre entre llaves como este:

```
{WXYZZZZ-6000-11D3-8CFE-0050048383C9}
```

De forma similar, Office 2003 tiene una entrada de registro como esta:

```
{WXYZZZZ-6000-11D3-8CFE-0150048383C9}
```

Donde:

**W:** Tipo de release = 9 (Fabricación)

**X:** Tipo de edición = 0 o 1

**YY:** SKU del producto

Office XP: 11 (Pro), 12 (estándar), 13 (Sm Bus), 28 (Pro w/ FrontPage), ...

Office 2003: 11 (Pro), 12 (Std), 13 (básico), CA (Sm Bus), ...

**ZZZZ:** Identificador de idioma hexadecimal del producto; inglés = 0409 (1033 decimal)

### Office XP SP3

El patrón de números que identifica Office XP SP3 se codifica en la siguiente expresión de relevancia:

```
Q: exists key whose (value "DisplayVersion" of it as string as version >= "10.0.6626.0" as version AND (character 1 of it = "9" AND (character 2 of it = "0" OR character 2 of it = "1") AND (first 2 of following text of first 3 of it = "11" OR first 2 of following text of first 3 of it = "12" OR first 2 of following text of first 3 of it = "13" OR first 2 of following text of first 3 of it = "28") AND (preceding text of first "}" of it ends with "6000-11D3-8CFE-0050048383C9")) of name of it) of key "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall" of registry
```

### Office 2003 SP2

El patrón de números que identifica a Office 2003 SP2 se codifica en la siguiente expresión de relevancia:

```
Q: exists key whose (value "DisplayVersion" of it as string as version >= "11.0.7969.0" as version AND (character 1 of it = "9" AND (character 2 of it = "0" OR character 2 of it = "1") AND (first 2 of following text of first 3 of it = "11" OR first 2 of following text of first 3 of it = "12" OR first 2 of following text of first 3 of it = "13" OR first 2 of following text of first 3 of it = "CA" OR first 2 of following text of first 3 of it = "E3") AND (preceding text of first "}" of it ends with "6000-11D3-8CFE-0150048383C9")) of name of it) of key "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall" of registry
```

## Detección de Service Packs de idioma extranjero

Normalmente, se trata de un problema sencillo de resolver para averiguar si un Service Pack de Windows está instalado si se inspecciona la versión Corrective Service Disk (CSD) del sistema operativo:

```
Q: csd version of operating system  
A: Service Pack 2
```

Por ejemplo, esta expresión de relevancia será verdadera si se ha instalado el Service Pack 1 o 2:

```
Q: (csd version of it = "Service Pack 1" or csd version of it = "Service Pack 2") of operating system  
A: True
```

Sin embargo, tanto el húngaro como el polaco tienen nombres no convencionales para sus Service Packs de Windows. Los siguientes inspectores identificarán debidamente estas anomalías:

### Húngaro

```
Q: (name of it = "WinXP" AND (it = "Service Pack 1" OR it = "Szervizcsomag 1" OR it = "Service Pack 2" OR it = "Szervizcsomag 2")) of csd version of it) of operating system  
Q: (name of it = "Win2003" AND (it = "" OR it = "Service Pack 1" OR it = "Szervizcsomag 1")) of csd version of it) of operating system
```

La primera expresión ha detectado el service pack en WinXP, y la segunda funciona para máquinas Win2000. Devolverán true si se ha instalado Service Pack 1 o 2.

## Polaco

Q: (name of it = "WinXP" AND (csd version of it = "Service Pack 1" OR csd version of it = "Service Pack 2" OR (csd version of it as lowercase starts with "dodatek" AND (csd version of it ends with " 1" OR csd version of it ends with " 2"))))  
of operating system

## Deconstrucción de XML

Puede deconstruir archivos XML utilizando los inspectores adecuados. Para este ejemplo, se da por supuesto que tiene un archivo XML denominado "c:\sample.xml" que tiene este aspecto:

```
<?xml version="1.0"?>
<message>
  <to>jim@rocket_science.com
    <name>Jim Neutron</name>
    <nickname>Jimmy the Geek</nickname>
  </to>
  <to>bob@rocket_science.com
    <name>Bob Goddard</name>
    <nickname>The Bobster</nickname>
  </to>
  <from>joe@big_sky.com</from>
  <subject>Let's do launch!</subject>
  <text>We are ready to test the new orbiter!</text>
</message>
```

A continuación se muestran los resultados de utilizar varios inspectores XML en este archivo:

Q: node names of child nodes of xml document of file "c:\sample.xml"

A: xml

A: message

Q: node names of child nodes of selects "message" of xml document of file "c:\sample.xml"

A: to

A: to

A: from

A: subject

A: text

Q: node names of child nodes of selects "message/to" of xml document of file "c:\sample.xml"

A: #text

A: name

A: nickname

A: #text

A: name

A: nickname

Q: unique values of node names of child nodes of selects "message/to" of xml document of file "c:\sample.xml"

A: #text

A: name

A: nickname

Q: node values of child nodes 0 of selects "message/to" of xml document of file "c:\sample.xml"

A: jim@rocket\_science.com%0a

A: bob@rocket\_science.com%0a

Q: node values of child nodes of selects "message/to/nickname" of xml document of file "c:\sample.xml"

A: Jimmy the Geek

A: The Bobster

Q: node names of next siblings of selects "message/to" of xml document of file "c:\sample.xml"

A: to

A: from

Q: node names of parent nodes of selects "message/to" of xml document of file "c:\sample.xml"

A: message

A: message

## Utilización de listas blancas

Puede utilizar expresiones de relevancia para buscar los equipos cliente para aplicaciones aprobadas. En primer lugar, cree un archivo denominado 'whitelist.txt' con los nombres de las aplicaciones aprobadas. Los nombres deben coincidir con el nombre de la aplicación registrada tal como lo devuelve el inspector regapp. A continuación se muestra una declaración de relevancia que destaca los nombres de las aplicaciones registradas:

Q: unique values of names of regapps

A: AcroRd32.exe

A: Acrobat Elements.exe

A: Ahqrun.exe

A: CTDVDA.exe

A: CTDVDET.exe

A: CTRegSvr.exe

A: EPSONCD.exe

A: EXCEL.EXE...

Ahora, cree un archivo de lista blanca con un nombre de aplicación por línea, como el siguiente:

Q: lines of file "c:\whitelist.txt"

A: acrodist.exe

A: conf.exe

A: EXCEL.EXE

A: IEXPLORE.EXE

A: msconfig.exe

A: OUTLOOK.EXE

A: Photoshop.exe

A: WINWORD.EXE

A: WINZIP32.EXE

A: wmpplayer.exe

A: wordpad.exe...

Ahora puede crear una expresión que compara las aplicaciones de lista blanca con las aplicaciones instaladas almacenadas en el registro. Esta expresión destaca una lista de las aplicaciones aprobadas que existen en el sistema cliente:

Q: (lines of file "c:/whitelist.txt", unique values of names of regapps) whose (item 0 of it is item 1 of it)

A: conf.exe, conf.exe

A: EXCEL.EXE, EXCEL.EXE

A: IEXPLORE.EXE, IEXPLORE.EXE

A: msconfig.exe, msconfig.exe

A: OUTLOOK.EXE, OUTLOOK.EXE

A: Photoshop.exe, Photoshop.exe

A: WINWORD.EXE, WINWORD.EXE

A: WINZIP32.EXE, WINZIP32.EXE

A: wmpplayer.exe, wmpplayer.exe

A: wordpad.exe, wordpad.exe

Puede probar los archivos que *no* están aprobados asegurándose de que una aplicación registrada determinada no exista en ningún lugar de la lista blanca. Se lleva a cabo comprobando la multiplicidad de las no coincidencias. Si las no coincidencias son iguales al número de líneas de la lista blanca, la aplicación no está en ningún lugar de la lista:

```
Q: unique values whose (multiplicity of it = number of lines of file
"c:/whitelist.txt") of (item 1 of it) of it whose ((item 1 of it)
does not start with (item 0 of it)) of ((lines of file "c:/whitelist.txt"),
unique values of names of regapps)
```

```
A: AHQTbU.exe
```

```
A: AcroRd32.exe
```

```
A: Ahqrun.exe
```

```
A: AudioCvt.exe
```

```
A: AudioHQU.exe
```

```
A: BrainExplorer.exe
```

```
A: CISDS.ds
```

```
A: CTCMSGo.exe
```

```
A: CTDVDA.exe ...
```

Ello produce una lista de aplicaciones en el sistema cliente que no están aprobadas. Esta lista puede controlar directamente una acción, o puede enviarse al administrador de IBM Endpoint Manager, quien puede decidir cómo manejarla.

Tenga en cuenta que también podría tener una lista negra que podría servir para identificar aplicaciones no aprobadas conocidas.

---

## Apéndice A. Gramática de lenguaje de relevancia

La gramática para el lenguaje de relevancia se puede expresar en el formato Backus Naur tal como se indica a continuación:

`<primary> := ( <expression> ) | string | numeral | it`

`<index> := phrase <primary> | phrase | <primary>`

`<property> := phrase <primary> <whose primary> <of property> |`

`primary <whose primary> <of property>`

`<cast> := <cast> as phrase | < property >`

`<phrase> := item | number | <expression>`

`<unary> := exists <unary> | notExists <unary> | not <unary> | - <unary> | <cast>`

`<productOperator> := * | / | mod | &`

`<product> := <product> <productOperator> <unary> | <unary>`

`<sum> := <sum> + <product> | <sum> - <product> | <product>`

`<relationExpr> := <sum> relation <sum> | <sum>`

`<relation> := relationOperator | relationPhrase`

`<relationOperator> := = | != | < | > | <= | >=`

`<relationPhrase> := is | is equal to | equals | is not | is not equal to | does not equal |`

`is greater than | is not greater than | is less than | is not less than |`

`is less than or equal to | is not less than or equal to |`

`is greater than or equal to | is not greater than or equal to | contains |`

`does not contain | is contained by | is not contained by | starts with |`

`does not start with | ends with | does not end with`

`<andExpression> := <andExpression> and <relationExpr> | <relationExpr>`

`<orExpression> := <orExpression> or <andExpression> | <andExpression>`

`<tuple> := <orExpression> , <tuple> | <orExpression>`

`<collection> := <collection> ; <tuple> | <tuple>`

<expression> := if <expression> then <expression> else <expression> |  
<collection>

## Operadores de relevancia

Operador	Efecto	Valor gramático
&	El operador de concatenación de la cadena de caracteres.	&
,	El operador de la tupla. Crea una tupla de objetos.	,
;	El operador de recopilación. Recopila sus operandos en un resultado plural.	;
+	El operador de suma.	+
-	El operador de resta.	-
*	El operador de multiplicación.	*
/	El operador de división.	/
=	Equivalente a la palabra clave 'es'.	relación
!=	Equivalente a 'no es'.	relación
<	El operador 'menor que'.	relación
<=	El operador 'menor que o igual a'.	relación
>	El operador 'mayor que'.	relación
>=	El operador 'mayor que o igual a'.	relación

## Precedencia y asociatividad

En el lenguaje de relevancia, el operador precedence es bastante estándar, p.ej. la multiplicación tiene una prioridad mayor que la suma, por lo que  $3+5*2 = 3+(5*2)$ , no  $(3+5)*2$ :

Q:  $3+5*2$   
A: 13

Los paréntesis, como era de esperar, prevalecen sobre el resto de los operadores:

Q:  $(3+5)*2$   
A: 16

Si dos operadores con la misma precedencia actúan en el mismo objeto, se llevará a cabo una opción para asociarse en primer lugar con el objeto a la izquierda o a la derecha. La suma y la resta se asocian a la izquierda, por lo tanto,  $1+2-3+4$  se procesa como  $((1+2)-3)+4$ .

La conversión también se asocia a la izquierda, por lo que '3 as string as integer' se interpreta como (3 as string) as integer:

Q: 3 as string as integer  
A: 3  
I: singular integer

A continuación se muestra una lista de los elementos del lenguaje, desde la procedencia más alta a la más baja, incluida la asociatividad donde sea adecuado:

Descripción	Valor gramático	Asociatividad
paréntesis	( )	
operador de conversión	as	left
operador unario	exists, not exists, not, -	
productos	*, /, mod, &	left
suma	+, -	left
relaciones	=, !=, <, <=, >, >=	
AND	y	left
OR	o	left
Tupla	,	
plural	;	left

Tenga en cuenta que no hay ninguna asociatividad listada para una relación, puesto que varios operadores de relación no pueden aparecer en la misma subexpresión. Por ejemplo:

Q: 1 is 1 is 1

A: This expression could not be parsed.

Asimismo, el operador tuple (coma) se asocia a la derecha, pero no aparece listado de esta forma en la tabla porque los paréntesis pueden cambiar la asociación. Por ejemplo, la primera expresión que aparece a continuación es un triple, pero la segunda es un par:

Q: (1), (2), (3)

A: 1, 2, 3

Q: (1), (2,(3))

A: 1, ( 2, 3 )

## Frases de clave de relevancia

En este apartado se presenta una tabla alfabética de las palabras clave en el lenguaje de relevancia, junto con sus valores gramaticales.

Palabra clave	Efecto	Valor gramático
a	Ignorado por el evaluador de relevancia. Se utiliza para mejorar la legibilidad.	<ninguno>
an	Ignorado por el evaluador de relevancia. Se utiliza para mejorar la legibilidad.	<ninguno>
and	El operador AND lógico. No evalúa el lado derecho si el lado izquierdo es falso.	y
as	El operador typecast, utilizado para convertir un tipo a otro.	as

Palabra clave	Efecto	Valor gramático
contains	Devuelve TRUE cuando una cadena de caracteres contiene otra cadena de caracteres como una subcadena de caracteres.	relación
does not contain	Equivalente a 'no contiene'.	relación
no termina con	Devuelve TRUE cuando una cadena de caracteres no termina con la subcadena de caracteres especificada.	relación
no es igual	Equivalente a 'no es'.	relación
no empieza por	Devuelve TRUE cuando una cadena de caracteres no empieza por la subcadena de caracteres especificada.	relación
else	Denota la vía de acceso alternativa en una sentencia 'if-then-else'.	else
acaba en	Devuelve TRUE cuando una cadena de caracteres termina con la subcadena de caracteres especificada.	relación
es igual a	Equivalente a 'es'.	relación
existe	Devuelve un booleano TRUE / FALSE que indica si existe un objeto.	existe
exist no	Equivalente a 'no existe'.	no existe
exists	Equivalente a 'existe'.	existe
exists no	Equivalente a 'no existe'.	no existe
if	La palabra clave para comenzar una expresión 'if-then-else'.	if
is	Devuelve TRUE cuando dos objetos son iguales. Tenga en cuenta que no todos los objetos se pueden probar para su igualdad. Equivalente al operador '='.	relación
se incluye en	Devuelve TRUE cuando una cadena de caracteres contiene otra cadena de caracteres como una subcadena de caracteres.	relación
es igual a	Equivalente a 'es'.	relación
es mayor que	La comparación '>'.	relación
es mayor o igual que	La comparación '>='.	relación
es menor que	La comparación '<'.	relación
es menor o igual que	La comparación '<='.	relación

Palabra clave	Efecto	Valor gramático
no es	Devuelve TRUE cuando dos objetos no son iguales. Tenga en cuenta que no todos los objetos se pueden comparar con esta palabra clave.	relación
no está contenido por	Devuelve TRUE cuando una cadena de caracteres no contiene otra cadena de caracteres como una subcadena de caracteres.	relación
no es igual a	Equivalente a la palabra clave 'no es' y al operador '!= '.	relación
no es mayor que	Equivalente a es menor que o igual a o '<='.	relación
no es mayor que o igual a	Equivalente a es menor que o '<'.	relación
no es menor que	Equivalente a es mayor que o igual que o '>='.	relación
no es menor o igual que	Equivalente a es mayor que o '>'.	relación
it	Una referencia al objeto directo más próximo o a la cláusula 'whose'.	it
item	Se utiliza para indexar en una tupla. Siempre devuelve un valor singular.	frase
items	Equivalente a elemento, pero devuelve un valor plural.	frase
mod	El operador del módulo.	mod
not	El operador lógico NOT.	relación
number	Devuelve el número de resultados en una expresión.	frase
of	Se utiliza para acceder a una propiedad de un objeto.	of
or	El operador lógico OR. No evalúa el lado derecho si el lado izquierdo es verdadero.	o
empieza con	Devuelve TRUE cuando una cadena de caracteres empieza con la subcadena de caracteres especificada.	relación
the	Ignorado por el evaluador de relevancia. Se utiliza para mejorar la legibilidad.	<ninguno>
then	Denota la vía de acceso principal que se debe emprender en una expresión if-then-else.	entonces
there do not exist	Equivalente a 'no existe'.	no existe

Palabra clave	Efecto	Valor gramático
there does not exist	Equivalente a 'no existe'.	no existe
there exist	Equivalente a 'existe'.	existe
there exist no	Equivalente a 'no existe'.	no existe
there exists	Equivalente a 'existe'.	existe
there exists no	Equivalente a 'no existe'.	no existe
whose	Se utiliza junto con la palabra clave 'it' para filtrar los resultados plurales.	whose

---

## Historial del lenguaje

### BES 1.x

- Incluye las primitivas de inicio, como los tipos booleanos, enteros, hercios, cadenas de caracteres, cuerda y de tiempo.
- Incluye inspectores de archivo, aplicación, versión, carpeta, unidad, sistema operativo, bios, registro, entorno, usuario actual.
- Incluye los inspectores del procesador, el dispositivo y la RAM.
- Incluye el servicio, la interfaz de red y los tipos de dirección IPV4.
- Incluye las propiedades de la acción.
- Incluye los inspectores de DMI.
- Incluye los operadores de conversión.

### BES 2.x

- Añade adaptadores de red.
- Añade Clientes y Valores.
- Añade WMI.

### BES 3.x

- Añade algunos inspectores de mundo, cadena de caracteres, cliente y servicio.

### BES 4.x

- Añade entradas del control de acceso y listas (ACE, ACL).
- Amplía las propiedades de la versión de archivos.
- Añade los números de coma flotante y los rangos de enteros.
- Añade inspectores de metabase.
- Añade recursos compartidos de red y correlación de puertos.
- Añade descriptores e identificaciones de seguridad.
- Presenta inspectores del cortafuegos de Internet.

### BES 5.0

- Añade conexiones.
- Añade inspectores de Fixlet.
- Añade inspectores de tipo de soporte.

### **BES 5.1**

- Las cláusulas if-then-else obtienen la capacidad de salvaguardarse contra el vocabulario desconocido enlazando a posteriori, lo que ayuda a grabar expresiones entre plataformas seguras.
- Presenta introspectores que pueden consultar el propio lenguaje.
- Presenta conjuntos de bits.
- Añade líneas de archivos y descendientes de carpetas.
- Añade más inspectores de cortafuegos.
- Añade el estado wake-on-lan.
- Añade inspectores XML.

### **BES 6.0**

- Presenta los inspectores de la sesión, incluidas las acciones BES, los sistemas, los sitios, los Fixlets, los resultados, las propiedades, los usuarios y los asistentes.
- Presenta inspectores de estadística.
- Presenta multiplicidad de series y enteros.
- Presenta resultados de varios valores (tuples).
- Añade la conjunción y la disyunción de listas booleanas.
- Presenta inspectores de nueva fecha, incluidos meses, años y el día de la semana, el mes y el año.
- Añade expresiones regulares.
- Amplía los inspectores de la lista de control de acceso (ACL).
- Mejora los inspectores de jerarquía de WMI.
- Añade los inspectores de creación de HTML.
- Añade los registros de eventos de Windows y los grupos locales.
- Presenta resúmenes del uso de la aplicación.

### **TEM 7.0**

- Añade nuevos inspectores de la sesión.
- Añade inspectores de nombre distinguido.
- Añade inspectores de cortafuegos.
- Añade inspectores de conjunto entero.
- Añade inspectores de conversión y de dirección IPv6

### **TEM 8.0**

- Presenta los inspectores de tarea planificada y del administrador de tareas.
- Añade búsquedas de comodín para archivos y carpetas.
- Añade inspectores de subcadena indexados.
- Añade operadores de rango.
- Presenta los inspectores de suma de tiempo.
- Presenta la suma y el producto de operadores en listas de coma flotante.
- Presenta nuevos operadores del sistema operativo e inspectores de variable del entorno.

## Mensajes de error

### "It" utilizado fuera de la cláusula "whose"

Se trata de un mensaje de error potencialmente confuso porque es perfectamente legal utilizar **it** sin una cláusula **whose** siempre que forme la sintaxis correctamente. Este mensaje sólo significa que el intérprete no sabe a qué se refiere **it**, normalmente debido a un error de sintaxis. Por ejemplo:

Q: it  
E: "It" used outside of "whose" clause.

Aquí la palabra clave "it" se utiliza en un contexto donde no puede hacer referencia a nada. A continuación se muestra otro ejemplo:

Q: system folder (name of it & pathname of it)  
E: "It" used outside of "whose" clause.

El error indica que el intérprete no sabe lo que es **it**, porque la declaración se ha formulado de forma incorrecta. La declaración correcta sería:

(name of it & pathname of it) of system folder

Para asegurarse de que **it** apunta hacia un objeto, debe asegurarse de que incluye **of <object>** tras la declaración.

### Es necesaria una expresión booleana

Este mensaje de error se dará cuando una declaración requiera un valor booleano para evaluarse, pero en su lugar recibe un tipo distinto. El lenguaje de relevancia se graba de forma fuerte, y la falta de atención a la coincidencia del tipo correcto es el origen de muchos errores. Entre otras cosas, es necesario un tipo booleano tras las declaraciones **if** y **whose**. Por ejemplo:

Q: Names of files whose (version of it = "5") of system folder

Este ejemplo funciona porque el valor entre paréntesis es un booleano. De lo contrario, obtendrá este error, como en este ejemplo incorrecto:

Q: Names of files whose (version of it) of system folder  
E: A boolean expression is required.

Aquí, la sentencia entre paréntesis no se evalúa con un booleano, por lo que la expresión falla con un error.

La prueba para una declaración **if-then** también debe ser un booleano. Puede pensar que un número no cero se podría evaluar en true, pero los tipos se deben hacer coincidir cuidadosamente y, en el lenguaje de relevancia, un número es un número, y no un booleano:

Q: if 1 then "boo" else "hoo"  
E: A boolean expression is required.

Uno (1) es un número; utilice un booleano en su lugar:

Q: if 1=1 then "boo" else "hoo"  
A: boo

Que es lo mismo que

Q: if true then "boo" else "hoo"  
A: boo

En las mismas líneas, puede pensar en probar un archivo ejecutable como este:

Q: If regapp "besclient.exe" then version of regapp "besclient.exe" as string  
else "N/A"  
E: A boolean expression is required.

En su lugar, debe comprobar la *existencia* del archivo, que es booleana:

If exists regapp "besclient.exe" then version of regapp "besclient.exe" as string  
else "N/A"

## Es necesaria una expresión singular

Este mensaje de error puede ser resultado de intentar comparar dos listas o una lista en un objeto. En general, todas las comparaciones deben realizarse entre dos objetos singulares. Por ejemplo:

Q: Versions of files of folder "c:\temp2" = versions of files of folder "c:\temp"  
e: A singular expression is required.

Devuelve el error porque la comparación de dos listas no está definida. Esto dará como resultado un error incluso si ambas carpetas contienen exactamente los mismos archivos. De manera similar,

Versions of files of folder "c:\temp" = "4.5"

Proporciona el mismo error debido a que no puede comparar una lista en un solo valor. Obtendrá este valor incluso si sólo hay un archivo en la carpeta "c:\temp" cuya versión es 4.5. Si desea iterar a través de varias comparaciones, utilice la sentencia **whose**:

versions of files whose (version of it = "4.5") of folder "c:\temp"

También puede obtener este error al intentar operar en dos objetos con distinta pluralidad, como en este ejemplo donde intentamos añadir un número y una lista:

Q: 1 + (2;3)  
E: A singular expression is required.

## Una constante de tipo serie tenía una %-sequence incorrecta

Puede insertar caracteres en una cadena especificando un signo de porcentaje y, a continuación, el valor hexadecimal de ASCII del carácter. Los literales codificados con porcentaje se analizan para asegurarse de que tienen el formato %hh, donde h es un número hexadecimal. Los códigos incorrectos (números no hexadecimales) desencadenarán este error.

Cuando se utiliza un signo de porcentaje en una cadena, el motor de relevancia busca en los dos siguientes caracteres para ver si se corresponden con un valor hexadecimal ASCII. Por ejemplo:

Q: "html uses %3Cangle brackets%3E"  
A: html uses <angle brackets>  
Q: "This is the %22truth%22"  
A: This is the "truth"

Sin embargo, una cadena con un signo de porcentaje en ella devolverá este error si cualquiera de las siguientes condiciones es verdadera:

- Hay menos de dos caracteres en la cadena después del signo de porcentaje.
- Los dos caracteres que siguen al signo de porcentaje no son caracteres hexadecimales válidos (0-9, a-f, A-F).



- **como cadena** para la mayoría de otros contextos

Si existe la conversión, pero no devuelve el tipo esperado, obtendrá este error.

### Tipos incompatibles

Q: 3; "three"  
E: Incompatible types.

Algunas operaciones tienen varios argumentos que deben ser del mismo tipo, o al menos compartir un tipo de base. Este ejemplo utiliza el operador (;) plural, que requiere una lista de objetos con el mismo tipo. Este mismo ejemplo funciona correctamente, aunque con un operador tupla (,):

Q: 3, "three"  
A: 3, three

Otro ejemplo es con una sentencia **if/then/else**. Devolverá la expresión después de **then** o **else**, pero ambos deben devolver el mismo tipo de objeto. Por ejemplo:

Q: if exists regapp "Besconsole.exe" then version of regapp "Besconsole.exe" else "Not Installed"  
E: Incompatible types.

El error se genera porque la expresión **then** devolverá una versión, mientras que la expresión **else** devolverá una cadena de caracteres. Asegúrese de que las dos sentencias devuelven el mismo tipo convirtiendo la versión a una cadena de caracteres:

If exists regapp "Besconsole.exe" then version of regapp "Besconsole.exe" as string else "Not Installed"

### Error definido por el inspector

Algunos errores son específicos del inspector.

Q: total duration of application usage summaries "excel.exe"  
E: application usage summary inspector is disabled

Este es un error creado en el inspector de 'application usage summary'.

### Operador no válido (<&)

Una secuencia de signos de puntuación legales no se resuelve para ser un operador reconocido.

### No hay contexto del inspector

Determinados inspectores sólo los puede evaluar el cliente y, por lo tanto, no funcionarán en el depurador del Fixlet. Si intenta evaluar uno de ellos en el depurador del Fixlet, recibirá el contexto **Sin inspector**. Un ejemplo común es algo parecido a:

Q: pending restart  
E: No inspector context.

En general, con el fin de evaluar sentencias que devuelven este error, debe definirlos como propiedades recuperadas desde la consola de IBM Endpoint Manager.

### Frase demasiado larga

Hay un límite de longitud de 64 caracteres en los nombres de propiedades.

## La expresión singular hace referencia a un objeto no exclusivo

Este mensaje de error se produce al intentar consultar una propiedad singular de varios objetos. Las expresiones de relevancia requieren una coincidencia entre el número de elementos del argumento y la pluralidad del Inspector. Por ejemplo:

```
Q: character of "abc"  
A: a  
E: Singular expression refers to non-unique object.
```

Esta expresión devolverá un carácter único, como solicita el **carácter** del inspector singular. Sin embargo, los caracteres extra del argumento generarán el error. Para evitar este error, utilice una expresión plural:

```
Q: characters of "abc"  
A: a  
A: b  
A: c
```

Cada Inspector tiene una versión singular y plural. Se indican para cada Inspector en las guías correspondientes.

Para devolver un solo valor, debe consultar un único objeto.

## La expresión singular hace referencia a un objeto no existente

Este es uno de los mensajes de error más comunes generados. Normalmente se obtiene como resultado de la consulta de una propiedad de un objeto que no existe, o de la consulta de una propiedad no existente de un objeto. Por ejemplo:

```
Q: version of file "mslib.dll" of folder "c:\temp"  
E: Singular expression refers to nonexistent object.
```

```
Q: character of ""  
E: Singular expression refers to nonexistent object.
```

Las expresiones singulares requieren al menos un valor. Para evitar este error, utilice una expresión plural, como **caracteres**. No devolverá un valor, ya que no está disponible ninguno, pero tampoco generará un error.

Para esta expresión en concreto, el error se genera si cualquiera de lo siguiente es correcto:

- La carpeta "c:\temp" no existe.
- No hay ningún archivo denominado "misspelled.dll" en la carpeta "c:\temp".
- El archivo "misspelled.dll" ubicado en la carpeta "c:\temp" no tiene una versión.

Puede crear una expresión que no devolverá un error utilizando plurales:

```
Q: versions of files "mslib.dll" of folders "c:\temp"
```

Sin embargo, una idea mejor es comprobar si existe en primer lugar:

```
Q: if (version of file "mslib.dll" of folder "c:\temp") then (version of file "mslib.dll" of folder "c:\temp" as string) else "n/a"
```

## El operador "<operator>" no está definido

Este error se produce cuando la operación elegida es incorrecta o cuando no se puede aplicar a los tipos especificados.

```
Q: 3 + "four"  
E: The operator "plus" is not defined.
```

También recibirá este mensaje si utiliza una palabra que el intérprete de relevancia no reconoce. A continuación, se muestran algunos ejemplos:

Q: Exists executable "file\_name.exe" of system folder  
E: The operator "executable" is not defined.

Aquí, la palabra 'executable' no es un mandato válido en el lenguaje de relevancia.

El error también se genera cuando se consulta una propiedad no existente:

Q: Version of key "HKLM/Software" of registry  
E: The operator "version" is not defined.

Aquí, aunque el lenguaje de relevancia conoce la palabra 'version', no la reconoce como una propiedad válida de **clave de registro**, y por lo tanto generará este error.

## El Operador "String" no está definido

Este error se produce en la máquina cliente (no en el depurador) cuando no hay manera de convertir el resultado como una cadena de caracteres, de modo que no se puede representar. Por ejemplo:

set of (1;2;3)

Esta expresión hace referencia a un conjunto legítimo, pero el lenguaje de relevancia no sabe cómo representarlo en el depurador. Para ver el conjunto, utilice una sintaxis como la siguiente:

Q: elements of set of (1;2;3)  
A: 1  
A: 2  
A: 3

Tenga en cuenta, sin embargo, que aunque el cliente obtenga un error de cadena de caracteres, el depurador del Fixlet generará un error distinto:

Q: set of (1;2;3)  
E: This expression evaluates to an unrepresentable object of type "integer set"

## El índice de tupla <index> está fuera de rango

Se puede acceder a las tuplas mediante un índice, pero no se debe superar el rango de los datos:

Q: item 2 of (1,"a")  
E: The tuple index 2 is out of range.

Los elementos de la tupla se basan en cero. Aquí el elemento está fuera de rango; los valores legales son 0 y 1:

Q: item 1 of (1,"a")  
A: a

## Esta expresión contenía un carácter que no está permitido

Este mensaje de error se produce cuando el intérprete de relevancia ha detectado un carácter que no reconoce. Por ejemplo:

Q: {pathname of regapp "besclient.exe"}  
E: This expression contained a character which is not allowed.

Las llaves son señales válidas en el lenguaje de acción, pero no en relevancia. Para usarlas en el depurador, tiene que descartar las llaves:

Q: pathname of regapp "besclient.exe"  
A: C:\Program Files (x86)\BigFix Enterprise\BES Client\BESClient.exe

Asimismo:

Q: #

E: This expression contained a character which is not allowed.

Aquí, un carácter que no tiene ningún significado en el lenguaje se ha utilizado fuera de las constantes de cadena de caracteres.

### **Esta expresión contenía un índice de tupla que no era un literal entero**

Q: item (4-3) of (1,"a")

E: This expression contained a tuple index which was not an integer literal.

El cálculo de un índice en una tupla no está permitido. Debe utilizar un literal:

Q: item 1 of (1,"a")

A: a

### **Esta expresión no se ha podido analizar**

El primer paso de interpretación de una sentencia de relevancia es analizar la expresión en sus diversos

componentes. Este error viene como resultado de una anomalía del motor de análisis. Normalmente viene causado

por paréntesis sin cerrar o por errores de sintaxis que implican determinadas palabras reservadas utilizadas por

el motor de análisis. Las palabras reservadas son sentencias sintácticas como **of**, **and**, **equals**,

etc.

A continuación se muestran un par de ejemplos:

Q: = )

E: This expression could not be parsed.

Q: Name of (file whose (version of it = "2.6") of system folder

E: This expression could not be parsed.

La expresión:

Exists file "name" of or system folder

Devolverá el mismo mensaje de error debido al uso incorrecto de las palabras reservadas **of** y **or**. Este error es un genérico para las modalidades no gramaticales.

### **Esta expresión se evalúa en un objeto no representable de tipo "<type>"**

El motor de relevancia requiere un marco básico para una respuesta. A veces, puede existir una sentencia y se puede utilizar, pero no se puede representar en el contexto del depurador:

Q: Key "HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\DirectX" of registry

E: This expression evaluates to an unrepresentable object of type "registry key"

Aunque exista esta clave, no hay ninguna operación indicada y el motor de relevancia no sabe qué producir. Tenemos que ser más específicos:

Q: exists Key "HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\DirectX" of registry

A: True



```
Q: if (year of current date as integer < 2006) then "Still good" else error  
"Expired"  
E: User-defined error: Expired
```

La palabra clave **error** le permite establecer sus propios códigos de error definidos por el usuario e implementarlos en la cláusula **else** de una sentencia **if/then/else**.

### **Valor no convertido**

El lenguaje de relevancia intenta de forma implícita convertir los resultados en una cadena de caracteres, en un número o en un valor booleano. Este error indica que el valor no se ha podido convertir.

Consulte "La conversión tiene un tipo incorrecto" y "El operador 'string' no está definido" en otra parte de esta lista.

---

## Apéndice B. Glosario

**Administrador del sitio:** el único operador de consola de IBM Endpoint Manager con el derecho de crear nuevos operadores.

**Base de datos de IBM Endpoint Manager:** un componente del sistema que almacena datos sobre los sistemas individuales y los Fixlets. Las interacciones de IBM Endpoint Manager Server afectan principalmente a esta base de datos, que se ejecuta en SQL Server.

**BigFix Enterprise Suite (BES):** el nombre anterior para IBM Endpoint Manager.

**Cliente:** software instalado en cada sistema de red que se gestionará en el IBM Endpoint Manager. El Cliente accederá a una agrupación de Fixlets, comprobará el sistema instalado en busca de vulnerabilidades, y envía al servidor un mensaje cuando se produce tal situación. Anteriormente conocido como el Cliente de BES, ahora se conoce como el cliente de IBM Endpoint Manager, o simplemente Cliente.

**Consola:** un programa de gestión que proporciona una visión general del estado de todos los equipos con el Cliente instalado en la red, que identifica lo que podría ser vulnerable y ofrece posibles soluciones. Anteriormente conocido como la Consola de BES, ahora se conoce como la consola de IBM Endpoint Manager, o simplemente Consola.

**Contraseña de acción:** ver la contraseña de inicio de sesión.

**Contraseña de firma:** contraseña (especificada cuando se instaló el sistema de IBM Endpoint Manager) utilizada por un operador de consola para firmar una acción para el despliegue. Se denomina contraseña de *acción* en la interfaz de la consola.

**Derechos de administración:** los operadores normales de la consola pueden estar limitados a un grupo concreto de equipos. Estos límites representan los derechos de gestión de cada usuario. Sólo un administrador de sitio o un operador maestro pueden asignar derechos de gestión.

**Encabezado maestro:** archivos que contienen los parámetros del proceso de IBM Endpoint Manager, incluidos los URL que apuntan a donde está disponible el contenido del Fixlet de confianza. El cliente de IBM Endpoint Manager lleva contenido a la empresa basada en los encabezados maestros suscritos.

**Fixlet:** un mecanismo para detectar y describir una situación problemática en un sistema y que le proporciona una solución automática.

**IBM Endpoint Manager:** herramienta de mantenimiento preventiva para entornos empresariales que supervisa los equipos que pertenecen a redes para detectar y solucionar vulnerabilidades con unas sencillas pulsaciones del ratón.

**Lenguaje de relevancia:** el lenguaje en el que se graban las expresiones de relevancia. Las expresiones de relevancia consultan el equipo del cliente para determinar si se desea la reparación. La reparación puede consistir en un script de acción que también puede utilizar expresiones de relevancia, con lo que se asegura que el elemento que se está arreglando es el mismo que el elemento que se identificó inicialmente.

**Lenguaje de secuencias de mandatos de acción:** el lenguaje empleado para crear secuencias de mandatos de acción. Las acciones se pueden crear en distintos lenguajes de secuencias de mandatos, incluidos intérpretes de AppleScript y Unix.

**Operador maestro:** un operador de consola con derechos administrativos. Un operador maestro puede hacer casi todo lo que puede hacer un administrador de sitio, excepto crear nuevos operadores.

**Operador:** persona que opera la consola de IBM Endpoint Manager. Los operadores normales pueden implementar acciones de Fixlet y editar determinados parámetros de los equipos. Los operadores maestros tienen privilegios adicionales, como la capacidad de asignar derechos de administración a otros operadores.

**Retransmisor:** cliente que ejecuta un software de servidor especial. Los retransmisores reducen la carga de trabajo del servidor y la red, minimizando las descargas directas entre servidores y clientes y comprimiendo los datos precedentes en la cadena. Los clientes descubren automáticamente los retransmisores y eligen dinámicamente el mejor retransmisor con el que establecer una conexión. Anteriormente conocido como Retransmisor de BES, ahora se conoce como el Retransmisor de IBM Endpoint Manager, o simplemente Retransmisor.

**Servidor:** conjunto de aplicaciones que interactúan entre sí (servidor web, CGI-BIN y servidor de base de datos) que coordina el intercambio de información hacia y desde equipos individuales en el sistema IBM Endpoint Manager. Los procesos del servidor pueden estar alojados en un único equipo o segmentados para ejecutarse en varios equipos distintos o de forma que estén replicados en servidores redundantes. Anteriormente conocido como Servidor BES, ahora se conoce como servidor de IBM Endpoint Manager, o simplemente Servidor.

**Servidores de Fixlet:** servidores web que ofrecen suscripciones al sitio de Fixlet. Pueden ser internos de la red empresarial o externos a la red (si se permite el acceso web directo externo).

**Sitio de Fixlet:** un origen de confianza desde el que el cliente obtiene Fixlets.

**Sitio personalizado:** puede crear su propio contenido personalizado y alojarlo en un sitio personalizado. Esto sólo puede hacerlo un operador maestro al que se le hayan proporcionado derechos para crear contenido personalizado (utilice el programa Admin para asignar estos usuarios).

---

## Apéndice C. Soporte

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.

Es posible que en otros países IBM® no ofrezca los productos, los servicios o las características que se describen en este documento. Póngase en contacto con el representante local de IBM para obtener información sobre los productos y servicios disponibles actualmente en su área. 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 aplicaciones pendientes de patente que abarquen el tema 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  
Armonk, NY 10504-1785 EE.UU.

Para consultas sobre licencias en las que se solicite información sobre el juego de caracteres de doble byte (DBCS), póngase en contacto con el departamento de Propiedad intelectual de IBM de su país o envíe las consultas, por escrito, a:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japón

El siguiente párrafo no se aplica en el Reino Unido ni en ningún otro país en el que dichas disposiciones entren en conflicto con la legislación local:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN GARANTÍA DE NINGUNA CLASE, NI EXPLÍCITA NI IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A, LAS GARANTÍAS O CONDICIONES IMPLÍCITAS DE NO VULNERACIÓN DE DERECHOS, COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN DETERMINADO.

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. En cualquier momento, IBM puede realizar mejoras y/o cambios en el producto y/o programa descritos en esta publicación sin aviso previo.

Cualquier referencia incluida en esta información a sitios Web que no sean de IBM sólo se proporciona para su comodidad y en ningún modo constituye una aprobación de dichos sitios web. Los materiales de dichos sitios Web no forman parte de los materiales para este producto de IBM y el uso de dichos sitios web corre a cuenta y riesgo del cliente.

IBM puede utilizar o distribuir la información que se le proporcione en la forma que considere apropiada, 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 EE.UU.

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.

Cualquier referencia a datos de rendimiento en este documento se ha 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, algunas mediciones pueden haberse calculado por extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a su entorno específico.

La información relativa a productos que no son de IBM se ha obtenido de los proveedores de estos productos, sus anuncios publicados y otras fuentes públicamente disponibles. IBM no ha probado dichos productos y no puede confirmar la precisión del rendimiento, compatibilidad u otra clase de afirmaciones relacionadas con los productos que no son de IBM. Las preguntas sobre las posibilidades de los productos no IBM deben dirigirse a los proveedores de dichos productos.

Todas las declaraciones sobre la dirección o los propósitos futuros de IBM están sujetas a cambios o retirada sin previo aviso, y sólo representan metas 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 está destinada a la 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 empresariales 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 similitud con nombres y direcciones utilizados por una empresa real es puramente casual.

#### 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 en cualquier formato sin necesidad de efectuar ningún pago a IBM, con el fin de desarrollar, utilizar, comercializar o distribuir programas de aplicación que se ajusten a la interfaz de programación de aplicaciones para la plataforma operativa para la cual se han escrito los programas de aplicación. Estos ejemplos no se han probado de forma exhaustiva bajo todas las condiciones. Por lo tanto, IBM no puede garantizar ni dar por sentada la fiabilidad, la capacidad de servicio ni el funcionamiento de dichos programas. Los programas de ejemplo se proporcionan "TAL CUAL", sin garantía de ningún tipo. IBM no se responsabilizará de daño alguno derivado del uso de los programas de ejemplo.

Si está viendo esta información en copia software, es posible que las fotografías y las ilustraciones en color no aparezcan.

### **Marcas registradas**

IBM, el logotipo de IBM e [ibm.com](http://ibm.com) son marcas registradas o marcas comerciales registradas de International Business Machines Corp., registradas en una gran cantidad de jurisdicciones en todo el mundo. Otros productos y nombres de servicios pueden ser marcas registradas de IBM o de otras compañías. Para obtener una lista actualizada de las marcas registradas de IBM, consulte la sección "Copyright and trademark information" del sitio web [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

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

IT Infrastructure Library es una marca registrada de la Agencia Central de Informática y Telecomunicaciones, que ahora forma parte de la Oficina de Comercio Gubernamental.

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 de sus filiales en Estados Unidos y en otros países.

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

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas registradas de Microsoft Corporation en los 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 registradas de Oracle y/o sus subsidiarias.

Cell Broadband Engine es una marca registrada de Sony Computer Entertainment, Inc. en EE.UU. y 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.





Impreso en España