

HP Operations Agent

para los sistemas operativos Windows®, HP-UX, Solaris, Linux y AIX

Versión de software: 11.00

Guía de usuario

Fecha de publicación del documento: Octubre de 2010
Fecha de publicación del software: Octubre de 2010



Avisos legales

Garantía

Las únicas garantías para los productos y servicios de HP se establecen en los términos de garantía expresos que acompañan a dichos productos y servicios. Nada de lo que contiene este documento podrá interpretarse como garantía adicional. HP no asume responsabilidad alguna por los errores editoriales, técnicos u omisiones contenidos en el presente documento.

La información aquí contenida está sujeta a cambios sin previo aviso.

Leyenda de derechos restringidos

Software informático confidencial. Se requiere una licencia válida de HP para su posesión, uso o copia. De conformidad con FAR 12.211 y 12.212, se autoriza el uso del software informático comercial, de documentación del software informático y de datos técnicos para componentes comerciales al gobierno de EE.UU. bajo licencia comercial estándar del fabricante.

Avisos de copyright

© Copyright 2010 Hewlett-Packard Development Company, L.P.

Avisos de marcas registradas

Intel® e Itanium® son marcas comerciales de Intel Corporation en EE.UU. y otros países.

Microsoft®, Windows®, Windows® XP y Windows Vista® son marcas comerciales registradas en EE.UU. de Microsoft Corporation.

UNIX® es una marca comercial registrada de The Open Group.

Reconocimientos

Este producto incluye software criptográfico escrito por Eric Young (eay@cryptsoft.com).

Este producto incluye software desarrollado por OpenSSL Project para su uso en OpenSSL Toolkit (<http://www.openssl.org/>).

Este producto incluye software escrito por Tim Hudson (tjh@cryptsoft.com).

Este producto incluye software desarrollado por Apache Software Foundation (<http://www.apache.org/>).

Este producto incluye una interfaz de la biblioteca de compresión de uso general "zlib" con Copyright © 1995-2002 Jean-loup Gailly y Mark Adler.

Actualizaciones de la documentación

La página de título de este documento contiene la siguiente información identificativa:

- Número de versión del software, que indica la versión del software.
- Fecha de publicación del documento, que cambia cada vez que se actualiza el documento.
- Fecha de publicación del software, que indica la fecha de publicación de esta versión del software.

Para buscar actualizaciones recientes o para asegurarse de estar usando la edición más reciente de un documento, vaya a:

<http://h20230.www2.hp.com/selfsolve/manuals>

Este sitio requiere que el usuario se registre para obtener un HP Passport y que inicie sesión.

Para registrarse y obtener un ID de HP Passport, vaya a:

<http://h20229.www2.hp.com/passport-registration.html>

O haga clic en el vínculo **Nuevo registro de usuario** en la página de inicio de sesión de HP Passport.

Recibirá ediciones actualizadas o nuevas si se suscribe al servicio de soporte técnico del producto. Póngase en contacto con su representante de ventas HP para mayor información.

Soporte técnico

Visite el sitio web HP Software Support Online en:

www.hp.com/go/hpsoftwaresupport

Este sitio web proporciona información de contacto y detalles sobre los productos, servicios y soporte técnico que ofrece HP Software.

El soporte técnico en línea de HP Software permite al cliente solucionar los problemas por sí mismo. Ofrece una forma rápida y eficaz de acceder a las herramientas de soporte técnico interactivas necesarias para gestionar su negocio. Como valorado cliente de soporte técnico, se puede beneficiar del sitio web de soporte técnico para:

- Buscar documentos en la base de conocimiento que le puedan interesar
- Enviar y rastrear casos de soporte técnico y solicitudes de mejora
- Descargar revisiones de software
- Gestionar contratos de soporte técnico
- Buscar contactos de soporte técnico de HP
- Revisar la información sobre servicios disponibles
- Participar en conversaciones con otros clientes de software
- Investigar y registrarse en cursos de formación de software

La mayoría de las áreas de soporte técnico requieren que se registre como usuario de HP Passport y que inicie sesión. Algunas pueden requerir también un contrato de soporte técnico. Para registrarse y obtener un ID de usuario de HP Passport, vaya a:

<http://h20229.www2.hp.com/passport-registration.html>

Para obtener más información sobre los niveles de acceso, vaya a:

http://h20230.www2.hp.com/new_access_levels.jsp

Tabla de contenido

1	Introducción	11
	Mapa de documentación	12
	Documentación relacionada.	13
2	Administración de la recopilación de datos	15
	logglob	16
	logappl	16
	logproc	16
	logpcmd	16
	logdev	17
	logtran	17
	logls	17
	logindx	17
	Modificación del archivo parm	19
	Registro de métricas calculadas con la normalización basada en núcleo	39
	Detención de la recopilación de datos	41
	Reinicio de la recopilación de datos	41
	Horario de verano	42
	Cambio manual de la hora del sistema	42
	Control del espacio en disco usado por los archivos de registro	42
	Almacenamiento de datos.	44
3	Cómo trabajar con HP Operations Agent	47
	Persistencia de un objeto monitorizado	49
	Cambio del usuario predeterminado en Windows	53
	Cambio del usuario predeterminado en UNIX/Linux	55
	Cambie el usuario predeterminado de comandos	57
4	Uso del programa utility	59
	Ejemplo de uso del modo interactivo y el modo por lotes	61
	Ejemplo de uso de la interfaz de línea de comandos	64
	Valores iniciales	65
	Definiciones iniciales de aplicación del archivo parm.	66
	Detalles cronológicos	67
	Resúmenes	69
5	Comandos del programa utility	73

6	Uso del programa extract	95
	Sintaxis	96
	Cómo exportar datos	102
	Tareas de exportación de ejemplo	104
	Archivos de datos de exportación	104
	Sintaxis del archivo de plantilla de exportación	106
	Creación de un gráfico o informe personalizado	109
	Salida de datos exportados	110
	Notas sobre los formatos ASCII y datafile	110
	Notas sobre el formato binario	111
7	Comandos del programa extract	119
	weekdays	151
	weekly	152
	yearly	154
8	Uso del programa cpush	157
	Ver métrica en tiempo real	158
	Modificar una clase de métrica	158
	Ver Ayuda de métrica	159
	Ver datos de métrica resumidos	159
9	Alarmas de rendimiento	161
	Generador de alarmas	162
	Envío de capturas SNMP a Network Node Manager	162
	Envío de mensajes a Operations Manager	162
	Ejecución de acciones locales	163
	Errores de procesamiento de alarmas	164
	Análisis de datos históricos de alarmas	164
	Componentes de definiciones de alarma	165
	Referencia de sintaxis de alarma	166
	Convenciones	166
	Elementos comunes	167
	Instrucción ALARM	170
	Instrucción ALERT	174
	Instrucción EXEC	174
	Instrucción PRINT	176
	Instrucción IF	177
	Instrucción LOOP	178
	Instrucción INCLUDE	179
	Instrucción USE	180
	Instrucción VAR	182
	Instrucción ALIAS	182
	Instrucción SYMPTOM	183
	Ejemplos de definiciones de alarma	184
	Personalización de definiciones de alarma	186

10 Asesor del componente RTMA	189
Ejecución de la secuencia de asesor en múltiples sistemas	191
Convenciones de sintaxis	191
Comentarios	192
Condiciones	192
Constantes	193
Expresiones	193
Nombres de métrica en sintaxis del asesor	193
Printlist	194
Variables	195
Instrucciones de sintaxis de asesor	195
11 Uso de Performance Collection Component en Windows.	223
Tipos y clases de datos	224
Extracción de datos de archivos de registro	226
Atributos de archivo	227
Plantillas de archivo de exportación	230
Archivos de exportación predeterminados	230
Almacenamiento de datos de archivos de registro	237
Períodos de almacenamiento	237
Anexión de datos almacenados	237
Sugerencias de almacenamiento	238
Rango de datos que se van a analizar	239
Informe de análisis	239
Formato de los archivos de orígenes de datos	250
Creación de una recopilación de contadores de rendimiento	258
Administración de una recopilación de contadores de rendimiento	258
Administración de ECBM desde la línea de comandos	260
12 Introducción general de Integración de Orígenes de Datos (DSI)	263
Creación de la especificación de clase	264
Recopilación y archivo de datos	265
Uso de datos	265
13 Uso de integración de orígenes de datos.	267
Definición del formato de archivo de registro	268
Modo en que están organizados los archivos de registro	269
Creación del conjunto de archivos de registro	270
Pruebas del archivo especificación de clase y el proceso de registro (Opcional)	270
Archivo de datos en el conjunto de archivos de registro	271
Uso de los datos registrados	271
14 Referencia de especificación de clase DSI	273
Sintaxis de especificación de clase	274
Descripción de CLASE	275
CLASE	275
ETIQUETA	276
Parámetros INDEX BY, MAX INDEXES y ROLL BY	276

Control del tamaño del archivo de registro	282
RECORDS PER HOUR	283
CAPACITY	284
Descripciones de métrica	285
MÉTRICA	285
ETIQUETA	286
Método de resumen	287
PRECISION	288
TYPE TEXT LENGTH	288
Especificación de clase de ejemplo	289
15 Referencia del programa DSI	291
Compilador sdlcomp	291
Sintaxis del compilador	292
Salida de compilador de ejemplo	293
Archivos de configuración	295
Definición de alarmas en métrica DSI	295
Procesamiento de alarmas	296
Procesamiento de datos de dsilog	299
Pruebas del proceso de registro con Sdlgendata	299
Creación de un archivo de formato	302
Cambio de especificación de clase	303
Exportación de datos DSI	304
Ejemplo de uso del programa extract para exportar datos del archivo de registro DSI	304
Ver datos en Performance Manager	304
Administrar datos con sdlutil	305
Sintaxis	305
16 Ejemplos de integración de orígenes de datos (DSI)	307
Escritura de una secuencia de comandos dsilog	308
Registro de datos vmstat	309
Creación de un archivo de especificación de clase	309
Compilación de un archivo de especificación de clase	309
Inicio del proceso de registro dsilog	310
Acceso a los datos	311
Creación de un archivo de especificación de clase	312
Compilación de un archivo de especificación de clase	313
Inicio del proceso de registro DSI	314
Registro de datos sar desde varios archivos	315
Creación de archivos de especificación de clase	315
Compilación de archivos de especificación de clase	320
Inicio del proceso de registro DSI	320
Registro de datos sar para varias opciones	321
Registro del número de usuarios del sistema	327
17 Mensajes de error	329
Mensajes de error generales	345

18 Descripción del seguimiento de transacciones	347
Ventajas del seguimiento de transacciones	348
Vista del cliente de los tiempos de las transacciones	348
Datos de transacciones	348
Objetivos de nivel de servicio	349
Requisitos del procesamiento de pedidos en tiempo real	349
Preparación de la aplicación de procesamiento de pedidos	350
Directrices para usar ARM	351
19 Funcionamiento del seguimiento de transacciones	353
Compatibilidad con ARM 2.0.	354
Llamada <code>arm_complete_transaction</code>	355
Ejemplo de aplicaciones instrumentadas para ARM	355
Especificación de nombres de aplicaciones y transacciones	356
Demonio de seguimiento de transacciones (<code>ttd</code>)	357
Devolución del estado una llamada a la API de ARM	357
Archivo de configuración de transacciones (<code>ttd.conf</code>)	359
Adición de nuevas aplicaciones	359
Adición de nuevas transacciones	359
Cambio de los valores de <code>range</code> o <code>SLO</code>	360
Palabras clave de archivos de configuración	360
Formato del archivo de configuración	362
Ejemplos de archivos de configuración	363
Directrices	364
Sobrecarga de E/S de disco	364
Sobrecargade la CPU	365
Sobrecargade la memoria	365
20 Introducción a las transacciones	367
Antes de comenzar	367
Definición de los objetivos de nivel de servicio	368
Modificación del archivo <code>parm</code>	369
Recopilación de datos de transacciones	369
Personalización del archivo de configuración (opcional)	370
Alarmas	373
21 Mensajes de seguimiento de transacciones	375
22 Métricas de transacciones	377
23 Ejemplos de seguimiento de transacciones	379
Ejemplo 1 (ejemplo de pseudocódigo de procesamiento de pedidos)	381
Ejemplo 2	381
Ejemplo 3	381
Ejemplo 4	382

24 Funciones avanzadas	383
25 Bibliotecas de transacciones	387
Ejemplos de opción de compilador C por plataforma	394
Biblioteca ARM NOP	395
Ejemplos	395
Configurar una aplicación (arm_init)	396
Configurar una transacción con UDM	396
Agregar la métrica	396
Iniciar la instancia de transacción con correladores	398
Iniciar la instancia de transacción sin correladores	399
Actualizar los datos de instancia de transacción con UDM	399
Actualizar los datos de instancia de transacción sin UDM	400
Detener la instancia de transacción con actualización de métrica	400
Detener la instancia de transacción sin actualización de métrica	401
Usar transacción completada con UDM:	402
Usar transacción completada sin UDM:	402
26 Registro y rastreo	403
Configuración de la directiva de registro	404
Identificación de la aplicación	405
Establecimiento del tipo de rastreo	407
Introducción al archivo de configuración de rastreo	407
Creación del archivo de configuración	410
Habilitación del mecanismo de rastreo	417
Visualización de mensajes de rastreo	418
Filtrado de rastreos	422
27 Solución de problemas	425
Índice	429

1 Introducción

HP Operations Agent permite monitorizar un sistema recopilando métricas que indican el estado, rendimiento y disponibilidad de los elementos fundamentales del sistema. El recopilador de datos integrado de HP Operations Agent permitirá al usuario recopilar y registrar métricas de rendimiento de sistemas de su entorno.

HP Operations Agent utiliza archivos de registro para almacenar las métricas recopiladas y proporciona un mecanismo para generar mensajes de alerta cuando las métricas recopiladas no coinciden con los valores de umbral predefinidos. Se pueden ver los datos de métricas recopiladas con la ayuda de herramientas de análisis de datos como HP Reporter y HP Performance Manager. Si se instala y configura el agente para que funcione con el servidor de administración HP Operations Manager (HPOM), se puede monitorizar centralmente el estado y el rendimiento del nodo de agente desde la consola de HPOM.

Mapa de documentación

El mapa de documentación presenta una lista que incluye los principales documentos de HP Operations Agent. Este mapa ayuda a identificar un documento en particular.

Figura 1 Mapa de documentación de HP Operations Agent



Documentación relacionada

La documentación del usuario relativa a HP Operations Agent se encuentra en el directorio `paperdocs` del soporte multimedia del producto. Para buscar actualizaciones recientes o para asegurarse de estar usando la edición más reciente de un documento, vaya a:

<http://h20230.www2.hp.com/selfsolve/manuals>

Este sitio requiere que el usuario se registre para obtener un HP Passport y que inicie sesión. Para registrarse y obtener un ID de HP Passport, vaya a:

<http://h20229.www2.hp.com/passport-registration.html>

O haga clic en el vínculo **Nuevo registro de usuario** en la página de inicio de sesión de HP Passport.

Tabla 1 Documentación del usuario acerca de HP Operations Agent

Documento	Uso	Temas principales
Notas de la versión	Consulte este documento para obtener más información sobre la versión del producto, nuevas funciones y problemas conocidos.	<ul style="list-style-type: none">• Nuevas funciones• Mejoras• Correcciones• Problemas conocidos y limitaciones
Guía de conceptos	La Guía de conceptos ayudará al usuario a comprender el mecanismo de funcionamiento de HP Operations Agent en distintos entornos.	<ul style="list-style-type: none">• Introducción a HP Operations Agent• Principales componentes de HP Operations Agent
Guía de instalación	Con la ayuda de la Guía de instalación, el usuario podrá instalar HP Operations Agent en los entornos siguientes: <ul style="list-style-type: none">• En un servidor de administración de HPOM (para su uso en el entorno de administración distribuido administrado por HPOM)• En un servidor independiente (para recopilar métricas de rendimiento del sistema del servidor local para su uso con herramientas de análisis de datos externos, como HP Performance Manager)	<ul style="list-style-type: none">• Instalación de HP Operations Agent desde la consola de HPOM• Instalación manual de HP Operations Agent• Licencias

Tabla 1 Documentación del usuario acerca de HP Operations Agent

Documento	Uso	Temas principales
Guía de implementación	Use esta guía para implementar HP Operations Agent en el entorno de monitorización desde un servidor de administración central de HPOM.	<ul style="list-style-type: none">• Establecimiento de un canal de comunicación seguro entre el servidor de administración de HPOM y HP Operations Agent.• Configuración de HP Operations Agent para su funcionamiento en entornos de clúster de High Availability.• Administración de la configuración de HP Operations Agent de forma remota desde la consola de HPOM.
Guía de referencia	La Guía de referencia incluye una lista con todos los comandos, procesos y servicios disponibles en el nodo del HP Operations Agent.	<ul style="list-style-type: none">• Utilidades de línea de comando• Variables de configuración

2 Administración de la recopilación de datos

HP Operations Agent incluye un recopilador de datos con el objeto de recopilar y registrar los datos de rendimiento del sistema del sistema monitorizado. El programa recopilador de datos (**scope**) permite almacenar los datos recopilados en el sistema. Los datos almacenados se pueden ver y analizar con HP Performance Manager o HP Reporter.

El recopilador `scope` permite ejecutar las tareas siguientes en el sistema:

- Reunir datos de métricas que indican el estado y el rendimiento del sistema monitorizado
- Registrar los datos de métricas recopilados en distintos archivos de registro



`Scope` no registra los datos de NFS, aunque estos datos se pueden ver mediante HP GlancePlus en el sistema de archivos local.

El archivo de parámetros de configuración (el archivo `parm`) permite configurar el mecanismo de registro de datos predeterminado del recopilador `scope`. Al modificar los parámetros del archivo `parm`, se pueden controlar las propiedades siguientes del recopilador `scope`:

- Intervalo del registro de datos
- Tipos de datos
- Tamaño de los archivos de registro

Tras instalar HP Operations Agent en el nodo, hay que configurar el mecanismo de recopilación de datos de `scope` mediante la modificación del archivo `parm`.

Archivos del registro de recopilación

El recopilador de datos `scope` (`scopeux` en los nodos de UNIX y Linux; `scopent` en los nodos de Windows) recopila y resume las medidas de rendimiento de la utilización de los recursos del sistema y registra los datos en los siguientes archivos de registro, dependiendo de las clases de datos especificadas en la línea de registro (`log line`) del archivo `parm`:

- `logglob`
- `logappl`
- `logproc`
- `logpcmd`
- `logdev`
- `logtran`
- `logls`
- `logindx`



Las marcas de hora de los registros de los archivos de registro indican la hora inicial de la recopilación de datos. El concepto de los procesos interesantes es un filtro que ayuda a minimizar el volumen de los datos registrados y que se controla desde el archivo `parm`.

Scope no registra los datos de NFS, aunque estos datos se pueden ver con HP GlancePlus en el sistema de archivos local.

logglob

El archivo `logglob` contiene medidas de la información del uso de recursos de todo el sistema (global). El recopilador `scope` resume los datos globales y registra periódicamente los datos a intervalos especificados en el archivo `parm`.

logappl

El archivo `logappl` contiene medidas agrupadas de los procesos que se ejecutan en aplicaciones definidas en el archivo `parm`. El recopilador `scope` resume los datos de aplicaciones y registra periódicamente los datos a intervalos especificados en el archivo `parm`.

logproc

El recopilador `scope` identifica aquellos procesos que puedan resultar interesantes y, a continuación, registra las medidas agrupadas de los procesos identificados en el archivo `logproc`. Scope identifica los procesos en función de las condiciones siguientes:

- Comienzo de un proceso
- Fin de un proceso
- Detalles de la configuración especificados en el archivo `parm`

logpcmd

El archivo `logpcmd` contiene los detalles de las actividades en línea de comandos realizadas en los procesos registrados en el archivo `logproc`.



No se puede controlar el tamaño, la sustitución y el intervalo de registro en el archivo `logpcmd`.

El archivo `logpcmd` está almacenado en el siguiente directorio del nodo:

- En Windows: `%ovdatadir%\datafiles`
- En UNIX (y Linux): `/var/opt/perf/datafiles`

El archivo puede almacenar un máximo de 25 MB de datos. Cuando se inicia el mecanismo de recopilación de datos, el recopilador `scope` crea la primera instancia del archivo `logpcmd` con la extensión 0. Cuando el archivo `logpcmd0` alcanza el límite de 25 MB, `scope` crea la segunda instancia del archivo `logpcmd`, el archivo `logpcmd1`.

Si el archivo `logpcmd1` excede del límite de 25 MB, los datos comienzan a sustituirse a partir del archivo `logpcmd0`.

logdev

El archivo `logdev` contiene medidas del rendimiento de dispositivos individuales. El recopilador `scope` resume los datos del dispositivo y registra periódicamente los datos a intervalos especificados en el archivo `parm`.

logtran

El archivo `logdev` contiene medidas de los datos de transacción de ARM. El recopilador `scope` resume los datos de transacción y registra periódicamente los datos a intervalos especificados en el archivo `parm`. Para obtener más información sobre cómo recopilar datos de transacción, consulte [Descripción del seguimiento de transacciones](#) en la página 347.

logls

El archivo `logls` contiene información sobre los sistemas lógicos. El recopilador `scope` resume los datos lógicos del sistema y registra periódicamente los datos a intervalos especificados en el archivo `parm`.

El archivo `logls` sólo está disponible en HP Operations Agent para HPVM, Hyper-V Host, vSphere Management Assistant (vMA), zonas globales de Solaris y AIX-LPAR.

logindx

El archivo `logindx` contiene la información necesaria para acceder a los datos de otros archivos de registro.

Estado de scope

Además de los archivos de registro, se crean otros dos archivos cuando se inicia `scope`. Son el archivo `RUN` que reside en el directorio `/var/opt/perf/datafiles/` y el archivo `status.scope` que se encuentra en el directorio `/var/opt/perf/`.

El archivo `RUN` se crea para indicar que el proceso `scope` se está ejecutando. Si se suprime este archivo, `scope` finaliza.

El archivo `/var/opt/perf/status.scope` sirve como registro de estado/error para el proceso `scope`. Se agrega nueva información a este archivo cada vez que el recopilador `scope` se inicia, se detiene o cuando se encuentra una advertencia o error. Para ver la información de estado y error más reciente de `scope`, se utiliza el comando `perfstat -t`.

Archivo parm

El archivo `parm` es un archivo de texto que contiene las instrucciones que indican a `scope` que registre medidas específicas de rendimiento.

Durante la instalación inicial, HP Operations Agent coloca el archivo `parm` predeterminado en dos directorios diferentes:

- En Windows:

▶ En Windows, el archivo `parm` existe con la extensión `.Mac` (`parm.mwc`).

— `%ovinstalldir%\newconfig`

— `%ovdatadir%`

- En HP-UX, Solaris y Linux:

— `/opt/perf/newconfig`

— `/var/opt/perf`

- En AIX:

— `/usr/lpp/perf/newconfig`

— `/var/opt/perf`

El mecanismo de recopilación de datos de `scope` está controlado por la configuración del archivo `parm` ubicado en el directorio `%ovdatadir%` (para Windows) o `/var/opt/perf` (para UNIX o Linux).

Si se desea modificar el mecanismo de recopilación de datos predeterminado, hay que modificar la configuración del archivo `parm` ubicado en el directorio `%ovdatadir%` (para Windows) o `/var/opt/perf` (para UNIX o Linux).

Cuando actualiza HP Operations Agent en un nodo (desde una versión anterior de HP Performance Agent), este proceso actualiza la copia del archivo `parm` disponible en el directorio `newconfig`. El archivo `parm` que reside en el otro directorio permanece sin modificar y continúa rigiendo el mecanismo de recopilación de datos en el nodo. En efecto, este método permite mantener el mecanismo de recopilación de datos configurados incluso después de actualizar el producto. En cualquier momento después de la actualización del producto, se pueden comparar los parámetros de la configuración existente del archivo `parm` con la nueva versión del archivo `parm` disponible en el directorio `newconfig` y, a continuación, realizar los cambios necesarios.

El archivo `parm` se configura para recopilar una cantidad promedio de datos de archivo de registro. La cantidad máxima depende del sistema. Consulte la descripción del parámetro `size` en [Descripciones de los parámetros](#) en la página 23.

Modificación del archivo parm

Se puede modificar el archivo `parm` con cualquier procesador de texto o editor que pueda guardar el archivo en formato ASCII.

Cuando se modifica el archivo `parm` o se crea uno nuevo, se aplican las siguientes reglas y convenciones:

- Cualquier parámetro especificado invalida los valores predeterminados. Consulte el archivo `parm` disponible en el directorio `newconfig` para obtener información sobre los valores predeterminados.
- No es relevante el orden en el que se especifican los parámetros en el archivo `parm`.
- Si se especifica un parámetro más de una vez, la última instancia del parámetro es la que se hace efectiva.
- Los parámetros `file`, `user`, `group`, `cmd`, `argv1` y `or` deben seguir la instrucción de la aplicación a la que definen.
- Los parámetros de la aplicación deben enumerarse en orden para que se agregue un proceso a la aplicación la primera vez que coincida.
- Se pueden usar letras en mayúscula, en minúscula o una combinación de ambas para todos los comandos e instrucciones de parámetros.
- Se pueden usar espacios en blanco o comas para separar palabras clave en cada instrucción.
- Se pueden comentar los parámetros del archivo `parm`. Se ignorarán las líneas que comiencen con un código de comentario (`/*`) o un signo de almohadilla (`#`).

Después de modificar el archivo `parm`, hay que reiniciar el componente Componente Performance Collection para que los cambios surtan efecto. Para reiniciar Componente Performance Collection, ejecute el comando siguiente:

En Windows

```
%ovinstalldir%bin\ovpacmd REFRESH COL
```

En HP-UX, Linux o Solaris

```
/opt/perf/bin/ovpa -restart scope
```

En AIX

```
/usr/lpp/perf/bin/ovpa -restart scope
```

Si desea usar el componente Real-Time Metric Access (RTMA), deberá reiniciar también el proceso `perfd`:

En Windows

```
%ovinstalldir%bin\ovpacmd REFRESH RTMA
```

En HP-UX, Linux o Solaris

```
/opt/perf/bin/pctl -restart
```

En AIX

```
/usr/lpp/perf/bin/pctl -restart
```

Parámetros del archivo parm

Scope está controlado por parámetros específicos en el archivo de parámetros de la recopilación (`parm`) que hacen lo siguiente:

- Establecer el espacio máximo en disco para los archivos de registro de `scope` sin procesar.
- Especificar los tipos de datos que se van a registrar.
- Especificar el intervalo en el que deben registrarse los datos.
- Especificar los atributos de procesos y métricas que se van a registrar.
- Definir los tipos de datos de rendimiento que se van a recopilar y registrar.
- Especificar los conjuntos de aplicaciones que pueden definir los usuarios y que deberían monitorizarse. Una aplicación puede ser uno o más programas que se monitorizan como grupo.
- Especificar cuándo `scope` debería realizar actividades diarias de mantenimiento de archivos de registro para que no impacten en la disponibilidad del sistema.

Se pueden modificar estos parámetros para configurar `scope` con objeto de registrar los datos del rendimiento que coincidan con los requisitos del sistema monitorizado (consulte [Modificación del archivo parm](#) en la página 19).

`scope` utiliza los parámetros del archivo `parm` enumerados en la [Tabla 2](#) en la página 20. Algunos de estos parámetros están destinados para sistemas específicos, tal como se indica en la tabla. Para obtener descripciones detalladas de estos parámetros, consulte [Descripciones de los parámetros](#) en la página 23 y [Parámetros de definiciones de aplicación](#) en la página 32.

Tabla 2 Parámetros del archivo parm utilizados por scope

Parámetro	Valores u opciones
id	Identificador del sistema
log	<ul style="list-style-type: none"> • global • application [=prm] [=all] ([=prm] sólo en HP-UX) • process • device=disk, lvm, cpu, filesystem, all (lvm sólo en HP-UX) • transaction=correlator, resource (resource sólo en HP-UX) • logicalsystem (Para Solaris, se admite el sistema lógico en el entorno operativo Solaris 10 o posterior) <p>En AIX, el sistema lógico se admite sólo en el entorno global LPAR en AIX 5L V5.3 ML3 y posterior y en WPAR en AIX 6.1 TL2.</p> <p>Para habilitar el registro lpar, logicalsystems=lpar logicalsystems</p> <p>Para habilitar el registro wpar, logicalsystems=wpar</p> <p>Para habilitar tanto el registro lpar como el wpar, logicalsystems=lpar,wpar logicalsystems=wpar,lpar logicalsystems=all</p>
mainttime	hh:mm (formato de hora de 24 horas)
scopetransactions	on off
subprocinterval	valor en segundos (no en HP-UX)
javaarg NOTA: Sólo en UNIX/Linux.	true false
procthreshold (igual que threshold)	cpu= <i>porcentaje</i> disk= <i>velocidad</i> (no en Linux ni Windows) memory=nn (valores en MB) nonew nokilled shortlived
apptreshold	cpu= <i>porcentaje</i>
diskthreshold	util= <i>velocidad</i>

Tabla 2 Parámetros del archivo parm utilizados por scope

Parámetro	Valores u opciones
bynetifthreshold	iorate= <i>velocidad</i>
fsthreshold	util= <i>velocidad</i>
lvthreshold	iorate= <i>velocidad</i>
bycputhreshold	cpu= <i>porcentaje</i>
wait	cpu= <i>porcentaje</i> (sólo HP-UX) disk= <i>porcentaje</i> (sólo HP-UX) mem= <i>porcentaje</i> (sólo HP-UX) sem= <i>porcentaje</i> (sólo HP-UX) lan= <i>porcentaje</i> (sólo HP-UX)
application	<i>nombre de la aplicación</i>
file	<i>nombre de archivo [, ...]</i>
argv1	primer argumento del comando [,]
cmd	expresión regular de la línea de comandos
user	<i>nombre de inicio de sesión del usuario [,]</i>
group	<i>nombre de grupo [,]</i>
or	
priority	low value-high value (el rango varía en función de la plataforma)
size	(valores en MB) process= <i>nn</i> (el valor máximo es 4096) El valor máximo para todas las clases siguientes es 2048. global= <i>nn</i> application= <i>nn</i> device= <i>nn</i> transaction= <i>nn</i> logicalsystem= <i>nn</i>
days	global= <i>nn</i> (los valores se dan en días) application= <i>nn</i> process= <i>nn</i> device= <i>nn</i> transaction= <i>nn</i> logicalsystem= <i>nn</i>
maintweekday	Sun Mon Tue Wed Thu Fri Sat
collectioninterval	process= <i>ss</i> (valores en segundos) global= <i>ss</i>

Tabla 2 Parámetros del archivo parm utilizados por scope

Parámetro	Valores u opciones
gapapp	blank unassignedprocesses existingapplicationname other
Flush	ss (valores en segundos) 0 (deshabilita el vaciado de datos)
zone_app	true false (sólo en Solaris 10 y superior)
proccmd NOTA: Sólo en UNIX/Linux.	0 (deshabilita el registro de los comandos de proceso) nnnn (se refiere al valor numérico de la longitud de un comando de proceso. El valor máximo es 1024)
ignore_mt	true (métricas de CPU de los valores de informes de clase global normalizados en el número activo de núcleos del sistema) true (métricas de CPU de los valores de informes de clase global normalizados en el número activo de subprocesos de CPU del sistema) ineffective (el subproceso múltiple está desactivado)

Descripciones de los parámetros

A continuación se muestran las descripciones de cada uno de los parámetros del archivo parm.

- ID
- Log
- Thresholds
- scopetransactions
- subprocinterval
- gapapp
- wait
- Size
- Mainttime
- Days
- Maintweekday
- javaarg
- Flush
- zone_app
- proccmd

- `ignore_mt`

ID

El valor ID del sistema es una cadena de caracteres que identifica el sistema. El ID predeterminado asignado es el nombre de host del sistema. Si se desea modificar el ID predeterminado asignado, hay que asegurarse de que todos los sistemas tienen cadenas ID únicas. Este identificador se incluye en los archivos de registro para identificar el sistema en el que se han recopilado los datos. Se puede especificar un máximo de 39 caracteres.

Log

El parámetro `log` especifica los tipos de datos que `scope` va a recopilar.

- `log global` permite a `scope` registrar registros globales en el archivo `logglob`. Es preciso tener registros de datos globales para ver y analizar datos de rendimiento en el sistema. Las métricas globales no se ven afectadas por las opciones de registro o valores de la aplicación o datos de proceso.
- `log application` permite a `scope` inscribir registros de aplicación activos en el archivo `logappl`. De manera predeterminada, `scope` registra sólo las aplicaciones que tienen procesos activos durante un intervalo.

- `log application=all` en el archivo `parm` permite a `scope` registrar todas las aplicaciones en el archivo `logappl` en cada intervalo, con independencia de si las aplicaciones están activas o no.

La opción `application=all` puede ser adecuada en circunstancias específicas en relación con el uso de alarmas de aplicación. Por ejemplo, se puede generar una alarma cuando una aplicación se vuelve inactiva (`APP_ALIVE_PROC`).

Si se habilita esta opción, el archivo de registro `logappl` aumenta de tamaño a una velocidad mayor, ya que todas las aplicaciones se registran en cada intervalo. Puede usar la función `scan` del programa `utility` para monitorizar la utilización de los archivos de registro `scope`.

- Sólo en HP-UX, puede especificar el parámetro `log application=prm` para permitir que `scope` registre los grupos de Process Resource Manager (PRM) en el archivo `logappl`. Si se especifica este parámetro, `scope` no registrará los conjuntos de aplicaciones definidas por el usuario enumerados en el archivo `parm`. Además, todas las métricas de aplicación recopiladas reflejarán un contexto de PRM y se agruparán por la métrica `APP_NAME_PRM_GROUPNAME`.

Las opciones de registro de aplicación no afectan a los datos globales o de proceso.

- `log process` permite a `scope` registrar información sobre procesos interesantes en el archivo `lobproc`. Un proceso puede ser interesante cuando se crea por primera vez, cuando finaliza y cuando excede de un umbral especificado en el archivo `parm` para una aplicación. Las opciones de registro del umbral de proceso no tienen efecto en los datos globales o de aplicación.
- `log device=disk,lvm,cpu,filesystem` permite a `scope` registrar la información sobre discos individuales, volúmenes lógicos (sólo HP-UX), CPU y sistemas de archivo en el archivo `logdev`.



No hay que utilizar `lvm` si el sistema monitorizado no se ejecuta con el sistema operativo HP-UX.

De manera predeterminada, sólo se registran los discos, volúmenes e interfaces cuya E/S se haya generado a través de ellos durante un intervalo. Los registros de `netif` (dispositivo LAN lógico) y los registros de discos (en HP-UX) siempre se registran con independencia de las opciones del dispositivo de registro seleccionado.

Por ejemplo, para solicitar el registro para discos individuales, volúmenes lógicos, CPU, interfaces de red, pero *no* sistemas de archivos individuales, hay que utilizar la configuración siguiente:

```
log device=disk,lvm,cpu.
```

- Cuando se especifica `filesystem`, todos los sistemas de archivos locales montados se registran en cada intervalo, con independencia de la actividad.
- `log device=all` en el archivo `parm` permite a `scope` registrar todos los dispositivos de discos, volúmenes lógicos, CPU e interfaz de red en el archivo `logappl` en cada intervalo, con independencia de si los dispositivos están activos o no.

Si se habilita esta opción, el archivo de registro `logdev` aumenta de tamaño a una velocidad mayor, ya que todos los dispositivos se registran en cada intervalo. Puede usar la función `scan` del programa `utility` para monitorizar la utilización de los archivos de registro y su cambio de tamaño.

- `log transaction` permite a `scope` registrar registros de transacción de ARM en el archivo `logtran`. Para permitir que `scope` recopile datos, debe ejecutarse en el sistema un proceso que esté instrumentado con la API de Application Response Measurement (ARM). (Para obtener más información, consulte [Descripción del seguimiento de transacciones](#) en la página 347.)

Los valores predeterminados para el parámetro `log transaction` son `no resource` y `no correlator`.

Para habilitar la recopilación de datos de recursos (sólo HP-UX) o recopilación de datos del correlador, hay que especificar **`log transaction=resource`** o **`log transaction=correlator`**. Ambos se pueden registrar especificando **`log transaction=resource, correlator`**.

- `log logicalsystems` permite a `scope` registrar información sobre los sistemas lógicos en el archivo `logls`. Los datos de los sistemas lógicos se resumen periódicamente a intervalos especificados en el archivo `parm`.

En AIX 6.1 TL2, el registro de BYLS para LPAR y WPAR se puede configurar usando el parámetro `logicalsystems` del archivo `parm`. Consulte [Tabla 2](#) en la página 20.

Los archivos de registro se crean de forma automática, con independencia de opciones de registro. Si se deshabilita un tipo particular de registro, el archivo de registro correspondiente no se suprimirá automáticamente del sistema monitorizado.

Si especifica `log` sin opciones, `scope` registra sólo los datos globales y de proceso.

Thresholds

Los parámetros de umbral permiten a `scope` registrar sólo la información crítica en los archivos de registro y descartar los detalles no necesarios y no críticos del sistema.

Los parámetros siguientes especifican los umbrales para distintas clases de métricas. Si el valor de umbral especificado excede para una instancia particular de una clase de datos, `scope` inscribe un registro para esa instancia.

Se pueden especificar valores inferiores para el umbral con objeto de permitir que `scope` registre más datos o se pueden especificar valores superiores para el umbral con el fin de que `scope` registre menos datos para que como promedio haya menos registros registrados. A continuación se enumeran los parámetros de umbral disponibles:

- `Procthreshold`
- `apptreshold`
- `diskthreshold`
- `bynetifthreshold`
- `fsthreshold`
- `lvthreshold`
- `bycputhreshold`

Procthreshold

El parámetro `procthreshold` se utiliza para establecer niveles de actividad con el fin de especificar criterios para procesos interesantes. Para usar este parámetro, hay que habilitar el registro del proceso. `procthreshold` sólo afecta a los procesos que están registrados y no afectan a otras clases de datos.

Hay que especificar opciones de umbral en la misma línea de parámetros (separadas por comas).

Opciones de `procthreshold` para datos de proceso

<code>cpu</code>	<p>Establece el porcentaje de utilización de CPU que debe exceder un proceso para convertirse en "interesante" y registrarse.</p> <p>El valor <code>percent</code> es un número real que indica el uso de CPU general. Por ejemplo, <code>cpu=7.5</code> indica que se registra un proceso si excede del 7,5 por ciento del uso de CPU en una muestra de 1 minuto.</p>
<code>disk</code>	<p>(No disponible en Linux o Windows.) Establece la velocidad de E/S del disco físico por segundo que debe exceder un proceso para convertirse en "interesante" y registrarse.</p> <p>El valor es un número real. Por ejemplo, <code>disk=8.0</code> indica que un proceso se registrará si la velocidad promedio de E/S del disco físico excede de 8 KB por segundos.</p>
<code>memory</code>	<p>Establece el umbral de memoria que debe exceder un proceso para convertirse en "interesante" y registrarse.</p> <p>El valor se especifica en unidades de megabyte y se redondea a los 100 KB más próximos. Si se establece, el umbral de memoria se compara con el valor de la métrica <code>PROC_MEM_VIRT</code>. Se registrarán los procesos que excedan del umbral de memoria, de manera similar a los umbrales de registro del proceso de CPU y de disco.</p>
<code>nonew</code>	<p>Deshabilita el registro de nuevos procesos si no han excedido ningún umbral. Si no se especifica, se registran todos los nuevos procesos. En HP-UX, si <code>shortlived</code> <i>no</i> se especifica, se registran sólo los nuevos procesos que duran más de un segundo.</p>

nokilled	Deshabilita el registro de los procesos cerrados si no han excedido ningún umbral. Si no se especifica, se registran todos los procesos cerrados. En HP-UX, si <i>shortlived no</i> se especifica, se registran sólo los procesos cerrados que duran más de un segundo.
shortlived	Permite el registro de procesos que se ejecutan durante menos de un segundo en un intervalo. (Lo que suele aumentar de manera importante el número de procesos registrados.) Si <i>scope</i> encuentra el parámetro <i>threshold shortlived</i> en el archivo <i>parm</i> , registra los procesos <i>shortlived</i> , con independencia de <i>cpu</i> o <i>disk threshold</i> , siempre que las opciones <i>nonew</i> y <i>nokilled</i> se supriman. El valor predeterminado es que no se registrará ningún proceso <i>shortlived</i> . (No hay que especificar <i>shortlived</i> en el parámetro <i>threshold</i> si no se desea registrar procesos <i>shortlived</i> .)
process	<i>procthreshold</i> especifica los umbrales para la clase PROCESS. Los valores predeterminados son los siguientes: <ul style="list-style-type: none"> • Procesos que han usado un valor superior al 10% de la CPU de un procesador durante el último intervalo • Procesos con un tamaño de memoria virtual establecido por encima de 900 MB. • Procesos con una velocidad promedio de E/S de disco físico superior a 5 KB por segundo

apptreshold

El parámetro *apptreshold* se utiliza para especificar valores de umbral para la clase de datos APPLICATION (métrica APP_CPU_TOTAL_UTIL). Los criterios de umbral se basan en el porcentaje de uso de CPU que debe exceder una aplicación para que ésta se registre en los archivos de registro.

La configuración predeterminada del archivo *parm* permite a *scope* registrar aplicaciones que usan más de 0% de CPU.

diskthreshold

El parámetro *apptreshold* se utiliza para especificar valores de umbral para la clase de datos DISK. Los criterios de umbral para la clase DISK se basan en el porcentaje de duración de tiempo en el que el disco ejecuta las E/S (métrica BYDSK_UTIL).

La configuración predeterminada del archivo *parm* permite que *scope* registre los detalles de los discos que están ocupados ejecutando E/S durante más del 10% de la duración de tiempo.

bynetifthreshold

El parámetro *bynetifthreshold* especifica los umbrales para la clase NETIF. Los criterios de umbral de la clase de datos NETIF se basan en el número de paquetes transferidos por la interfaz de red por segundo (métrica BYNETIF_PACKET_RATE).

La configuración predeterminada del archivo *parm* permite a *scope* registrar los detalles de las interfaces de red que transfieren más de 60 paquetes por segundo. Si no se especifica el valor de este parámetro o si el parámetro se ha convertido en comentario, *scope* registra los detalles de todas las interfaces de red que no están inactivas.

fsthreshold

El parámetro `fsthreshold` especifica los umbrales para la clase `FILESYSTEM`. Los criterios de umbral de la clase de datos `file system` están basados en el porcentaje de espacio de disco utilizado por las clases `filesystem` (métrica `FS_SPACE_UTIL`).

La configuración predeterminada del archivo `parm` permite a `scope` registrar los detalles de las clases `filesystem` que utilizan más del 70% de espacio de disco.

lvthreshold

El parámetro `lvthreshold` especifica los umbrales para la clase `LOGICALVOLUME`. Los valores de umbral de la clase de datos del volumen lógico están basados en las E/S por segundo (`LV_READ_RATE + LV_WRITE_RATE`).

La configuración predeterminada del archivo `parm` permite a `scope` registrar los detalles de los volúmenes lógicos que tienen más de 35 E/S por segundo.

bycputhreshold

El parámetro `bycputhreshold` especifica los umbrales para la clase `CPU`. Los criterios de umbrales de la clase de datos `CPU` se basan en el porcentaje de tiempo que está ocupada la `CPU` (`BYCPU_CPU_TOTAL_UTIL`).

La configuración predeterminada del archivo `parm` permite a `scope` registrar los detalles de las clases `CPU` que están ocupadas más del 90% del tiempo.

scopetransactions

El mismo compilador `scope` está instrumentado con las llamadas a la API de ARM (Application Response Measurement) para registrar sus propias transacciones. El indicador `scopetransactions` determina si se van a registrar o no las transacciones `scope`. El valor predeterminado es `scopetransactions=on`; `scope` registrará dos transacciones: `Scope_Get_Process_Metrics` y `Scope_Get_Global_Metrics`. Si no se desea registrar estas transacciones de `scope`, hay que especificar `scopetransactions=off`. Una tercera transacción, `Scope_Log_Headers`, siempre estará registrada; no se ve afectada por `scopetransactions=off`.

Para obtener más información sobre ARM, consulte [Descripción del seguimiento de transacciones](#) en la página 347.)

subprocinterval

El parámetro `subprocinterval`, si se especifica, invalida el valor predeterminado que utiliza `scope` para muestrear los datos del proceso. Los datos del proceso y los datos globales se registran periódicamente a intervalos especificados en el archivo `parm`. Sin embargo, `scope` sondea su instrumentación cada pocos segundos para capturar actividades de corto plazo. De manera predeterminada, el intervalo de muestreo de instrumentación es de 5 segundos. El intervalo de registro de los datos del proceso debe ser un múltiplo par de `subprocinterval`. Para obtener más información, consulte [Configuración de intervalos de registro de datos](#) en la página 37.

En algunos sistemas con miles de subprocesos o procesos activos, `subprocinterval` debería aumentarse para reducir la sobrecarga general de `scope`. En otros sistemas con varios procesos de corta duración que se desean registrar, hay que tener en cuenta la reducción de `subprocinterval`, aunque en este caso debería monitorizarse rigurosamente el efecto en la sobrecarga de `scope`. Esta configuración puede adoptar valores que son factores del intervalo de registro del proceso, tal como se especificó en el archivo `parm`.



Unos valores inferiores en `subprocinterval` reducirán el intervalo entre las métricas globales y la suma de aplicaciones en todos los sistemas operativos que no sean HP-UX.

gapapp

El parámetro `gapapp` del archivo `parm` controla la modificación de las clases de datos de la aplicación para que tengan en cuenta cualquier diferencia entre los datos globales (de todo el sistema) y la suma de los datos de aplicación.

Los datos de aplicación proceden de la instrumentación a nivel del proceso. Por lo general, existe una diferencia entre las métricas globales y la suma de aplicaciones. En aquellos sistemas con elevadas tasas de creación de procesos la diferencia puede ser significativa. Se pueden elegir las opciones siguientes:

- Si `gapapp` está en blanco, se agregará una aplicación denominada `gapapp` a la lista de aplicaciones.
- Si `gapapp = UnassignedProcesses`, se agregará una aplicación denominada `UnassignedProcesses` a la lista de aplicaciones.
- Si `gapapp = ExistingApplicationName (o) gapapp = other`, se agregará la diferencia de los valores globales a la aplicación especificada en lugar de ser registrada por separado y agregar una nueva entrada a la lista de aplicaciones.

wait

El parámetro `wait` (sólo HP-UX) se puede utilizar para capturar los detalles de los procesos que esperan recursos del sistema. Se puede especificar el valor del parámetro `wait` en porcentaje. Si un proceso espera los recursos del sistema: `cpu`, `disk`, `mem`, `sem` y `lan` con un porcentaje de intervalo superior al valor especificado para el parámetro `wait`, los detalles de dicho proceso se registran en el archivo `logproc`.

Consulte [Parámetros del archivo parm utilizados por scope](#) para obtener información sobre los valores y opciones.

Por ejemplo, si el intervalo de registro del proceso se ha establecido en 60 segundos y el parámetro `wait` para la CPU se establece en el 50%, cualquier proceso que espere por la CPU durante más o igual que 30 segundos se captura en el archivo `logproc`.

Size

El parámetro `size` se utiliza para establecer el tamaño máximo (en megabytes) de cualquier archivo de registro sin procesar. No se puede establecer el tamaño para que sea inferior a un megabyte.

El recopilador `scope` lee estas especificaciones cuando se inicia. Si alguno de estos archivos de registro llega a su tamaño máximo durante la recopilación, continuará aumentando hasta llegar al valor de `mainttime`, momento en el que se revertirán automáticamente. Durante una reversión, el 25 por ciento de los datos más antiguos se suprimirán del archivo de registro. Los archivos de registro sin procesar sólo se designan para almacenar un máximo de datos de un año, si no están limitados por el parámetro `size`. Consulte [Resumen de contenidos del archivo de registro](#) en la página 71 y [Resumen de espacio libre del archivo](#) en la página 72 en la sección [Detalles del informe de exploración de utility](#) del capítulo 3.

Si se cambia la especificación `size` del archivo `parm`, `scope` lo detecta durante el inicio. Si el tamaño máximo del archivo de registro disminuye hasta el punto donde no se ajustan los datos existentes, tendrá un lugar un cambio de tamaño automático durante el inicio de `scope`. Si los datos existentes se ajustan al nuevo tamaño máximo especificado, no se realiza ninguna acción.

El comando `resize` crea el nuevo archivo `scopelog` en el directorio establecido por la variable de entorno `TMPDIR` antes de eliminar el archivo de registro original. Consulte [Instrucciones sobre su uso](#).

Cualquier cambio que realice en el tamaño máximo de un archivo de registro surtirá efecto en el momento especificado en el parámetro `mainttime`.



Con independencia de los parámetros `size`, el tamaño máximo de los archivos de registro `scope` también estarán limitados por la cantidad de datos almacenados en un año. Los archivos de registro sin procesar de `scope` no pueden contener más de un año de datos, por tanto si los registros se extienden más allá de esta duración, se sobrescribirán los datos más antiguos que un año. Consulte [extract](#) en la página 130 para obtener más información sobre cómo crear archivos de registro de almacenamiento si se desea disponer de más de un año de datos.

Mainttime

Si es necesario, `scope` revertirá los archivos de registro sólo en un período de tiempo específico cada día. El período de tiempo predeterminado se puede modificar con el parámetro `mainttime`. Por ejemplo, si se establece `mainttime=8:30`, se realizará el mantenimiento del archivo de registro a las 8:30 a.m. cada día.

Se recomienda establecer `mainttime` en un período de tiempo de baja utilización del sistema.



El mantenimiento del archivo de registro sólo descarta los datos más antiguos a un día, por tanto si un archivo de registro como `logproc` aumenta rápidamente y alcanza su límite en 24 horas, su tamaño puede exceder del límite de tamaño configurado.

Days

El parámetro `days` especifica el número máximo de días de los datos, que cualquier archivo de registro de datos sin procesar puede almacenar en un determinado punto de tiempo. El valor de esta parámetro debe encontrarse en el rango de 1 a 365. Este parámetro permite al recopilador de datos `scope` mantener los archivos de registro.

Durante la recopilación de datos, si el número de días de datos del archivo de registro alcanza los días especificados en el parámetro `days`, la recopilación de datos continuará hasta que se alcance el día especificado en el parámetro `maintweekday`. Cuando se alcanza `maintweekday`, el archivo de registro se revertirá automáticamente a `mainttime`. Durante la reversión, los datos recopilados después del parámetro `days` que hayan alcanzado su valor máximo se suprimirán del archivo de registro.



Si los archivos de registro se reversionan durante la recopilación de datos o si se alcanza el valor especificado en el parámetro `size` un día específico antes del parámetro `days`, en este caso el parámetro `size` invalida el parámetro `days`.

Por ejemplo, si se utilizan `"size global=20"` y `"days global=40"` en el archivo `parm` y si los archivos de registro alcanzan el tamaño máximo de 20 MB 40 días antes de que se registren los datos en el archivo de registro, se realiza la reversión del archivo de registro en función del parámetro `size`.

Maintweekday

El parámetro `maintweekday` especifica el día de la semana en el que ocurre la reversión del archivo de registro si se alcanza el parámetro `days`. La reversión ocurrirá en el valor establecido en `maintime`.

Por ejemplo, si se utiliza “`maintweekday=Mon`” en el archivo `parm`, se realizará la reversión del archivo de registro cuando el valor especificado en el parámetro `days` llegue al lunes (Mon) en `maintime`. Se recomienda establecer el valor de `maintweekday` en un día de la semana con baja utilización del sistema.

- ▶ El parámetro `maintweekday` es opcional. Si se especifica el parámetro `maintweekday` en el archivo `parm`, debería utilizarse con el parámetro `days`. Este parámetro no se tendrá en cuenta si no se utiliza con el parámetro `days` del archivo `parm`. Si no se especifica `maintweekday` en el archivo `parm` aunque se especifique el parámetro `days`, el valor predeterminado será “`maintweekday=Sun`”.

Por ejemplo, si “`daysglobal=30`”, “`application=20`”, “`process=30`”, “`device=20`”, “`transaction=10`”, “`maintweekday=Wed`” y si el archivo de registro alcanza el número de días especificado en el parámetro `days`, la recopilación de datos continuará hasta el día especificado en `maintweekday`. Una vez alcanzado `maintweekday`, tendrá lugar la reversión del archivo de registro eliminando el número excedido de días de los datos desde el comienzo del archivo de registro. Este mantenimiento se realizará en `maintime`.

javaarg

- ▶ Este parámetro sólo es válido en UNIX/Linux.

El parámetro `javaarg` es un indicador que se puede establecer en `true` o en `false`. SÓLO afecta al valor de la métrica `proc_proc_argv1`.

Si `javaarg` se establece en `false` o no se define en el archivo `parm`, la métrica `proc_proc_argv1` siempre se establecerá en el valor del primer argumento de la cadena de comandos en el proceso.

Si `javaarg` se establece a `true`, la métrica `proc_proc_argv1` se invalida, sólo para los procesos Java, con la especificación `class` o `jar` si se puede encontrar en la cadena de comandos. En otras palabras, en aquellos procesos cuyos nombres de archivo sean `java` o `jre`, la métrica `proc_proc_argv1` se invalida con el primer argumento sin guión inicial no seguido por un `-classpath` o un `-cp`, asumiendo que los datos se pueden encontrar en la lista de argumentos proporcionada por el sistema operativo.

Aunque esto puede parecer complicado, es muy sencillo cuando se están ejecutando procesos java en el sistema: establecer **`javaarg=true`** y la métrica `proc_proc_argv1` se registra con el nombre `class` o `jar`. Este sistema puede resultar muy útil si se desea definir aplicaciones específicas para java. Cuando el nombre de clase se encuentra en `proc_proc_argv1`, se puede usar el calificador `argv1= application` para definir la aplicación por nombre de clase.

Flush

El parámetro `flush` especifica los intervalos de registro de datos (en segundos) en los que se registrarán todas las instancias de los datos de la aplicación y del dispositivo. Los intervalos de vaciado (`flush`) deben encontrarse entre 300 y 32700 y ser un múltiplo par de 300.

El valor predeterminado del intervalo de `flush` es de 3600 para todas las instancias de datos de aplicación y de dispositivo.

El parámetro `flush` se puede deshabilitar especificando el valor en 0 (cero). Si el parámetro `flush` se establece en 0, `scope` no registrará ningún dato de aplicación o dispositivo que no alcance los umbrales especificados en el archivo `parm`.

`zone_app`

El indicador `zone_app` permite a Componente Performance Collection recopilar datos específicos de zonas de Solaris. Si el indicador `zone_app` está establecido en `true`, afectará a la recopilación de todas las métricas de clase de aplicación (`APP_*`). Se ignorarán todos los conjuntos de aplicaciones definidos por el usuario del archivo `parm` y las métricas de aplicación se recopilarán en función de las zonas que se ejecutan en los equipos instalados de Componente Performance Collection.

Por ejemplo, considérese un equipo Solaris que se está ejecutando con dos zonas no globales: "zone1" y "zone2". Componente Performance Collection ignorará los conjuntos de aplicaciones del archivo `parm` y creará tres aplicaciones, llamadas "global", "zone1" y "zone2". La medida del rendimiento para cada aplicación se basará en los valores de medida obtenidos desde los procesos en ejecución en las zonas respectivas. Por ejemplo, la métrica `APP_CPU_TOTAL_UTIL` para la zona "zone1" se calculará basándose en los valores de utilización de la CPU para todos los procesos que se ejecutan en `zone1`.

Si el indicador `zone_app` no está habilitado o si Componente Performance Collection está ejecutándose en Solaris 9 o una versión anterior, la métrica `APP_LS_ID` informará un valor de No disponible (na). La agrupación de aplicaciones se realizará en función de los conjuntos de aplicaciones definidos por el usuario enumerados en el archivo `parm`.



Las zonas sólo se admiten en Solaris 10 y versiones superiores.

`proccmd`



Este parámetro sólo es válido en UNIX/Linux.

El parámetro `proccmd` permite el registro de los comandos de proceso en el almacén de datos de HP Operations Agent. Se puede especificar la longitud del comando de proceso como un valor numérico de este parámetro. El valor numérico máximo es 1024. De manera predeterminada, el valor de este parámetro se establece en 0 y se deshabilita el registro de comandos de proceso.

`ignore_mt`

Si se establece este parámetro en `true`, Componente Performance Collection registra todas las métricas relacionadas con la CPU de la clase global después de normalizar los valores de métricas en el número de núcleos activos en el sistema monitorizado.

Si se establece este parámetro en `false`, Componente Performance Collection registra todas las métricas relacionadas con la CPU de la clase global después de normalizar los valores de métricas en el número de subprocesos en el sistema monitorizado.

Componente Performance Collection ignora este parámetro si la propiedad de subprocesos múltiples está deshabilitada en el sistema. Como resultado, el valor de la métrica `GBL_IGNORE_MT` se registra como `true`.



Si se habilita o deshabilita el subprocesamiento múltiple simultáneo (SMT, Simultaneous Multi-Threading) en un sistema Windows, Linux o Solaris, hay que reiniciar el sistema.

Parámetros de definiciones de aplicación

Los parámetros siguientes pertenecen a las definiciones de aplicación: `application`, `file`, `user`, `group`, `cmd`, `argv1` y `or`.

Componente Performance Collection agrupa los procesos relacionados lógicamente en una aplicación para registrar el efecto combinado de los procesos en recursos informáticos, como memoria y CPU.

- ▶ En el modo PRM (sólo para HP-UX), se registran los grupos PRM activos y se ignoran los conjuntos de aplicación definidos por el usuario enumerados en el archivo `parm`.

Una aplicación puede ser una lista de archivos (nombre de programas básicos), una lista de comandos o una combinación de ellos, también calificados por nombres de usuarios, nombres de grupo o selecciones de argumentos. Todos estos calificadores de aplicación se pueden usar individualmente o con otros. Por ejemplo, si se utilizan los calificadores `cmd` y `user`, un proceso debe cumplir la especificación tanto de la cadena de comandos y del nombre de usuario para pertenecer a esa aplicación. A continuación se describe con detalle cada uno de los calificadores.

- ▶ Los procesos del sistema pertenecen a una única aplicación. Ningún proceso se incluye en dos o más aplicaciones.

Application

El nombre de aplicación define una aplicación o clase que agrupa varios procesos e informes en sus actividades combinadas.

- El nombre de aplicación es una cadena de hasta 10 caracteres que se utiliza para definir la aplicación.
- Los nombres de aplicación pueden ir en mayúsculas o minúsculas y contener letras, números, caracteres de subrayado y blancos incrustados. No hay que usar el mismo nombre de aplicación más de una vez en el archivo `parm`.
- Se puede utilizar de manera opcional un signo igual (=) entre la palabra clave `application` y el nombre de la aplicación.
- El parámetro `application` debe preceder a cualquier combinación de parámetros `file`, `user`, `group`, `cmd`, `argv1` y `or` que hagan referencia a él, y todos estos parámetros se aplicarán en la última definición de carga de trabajo de la aplicación.
- Cada parámetro puede constar de hasta 170 caracteres, incluyendo el carácter de retorno de carro, sin que se permitan caracteres de continuación. Si la lista de archivos es superior a 170 caracteres, hay que continuar la lista en la línea siguiente después de otra instrucción `file`, `user`, `group`, `cmd` o `argv1`.
- Se pueden definir hasta 998 aplicaciones. Componente Performance Collection predefine una aplicación denominada `other`. La aplicación `other` recopila todos los procesos no capturados por las instrucciones `application` en el archivo `parm`.

Por ejemplo:

```
application Prog_Dev
file vi,cc,ccom,pc,pascomp,dbx,xdb

application xyz
file xyz*,startxyz
```

Se puede tener un máximo de 4096 especificaciones `file`, `user`, `group`, `argv1` y `cmd` en todas las aplicaciones combinadas. En el ejemplo anterior se incluyen nueve especificaciones de archivo. (`xyz*` cuenta como una única especificación aunque pueda coincidir con más de un archivo de programa.)

Si se incluye un archivo de programa en más de una aplicación, se registra en la primera aplicación que lo contiene.

El archivo `parm` predeterminado contiene algunas aplicaciones de muestra que se pueden modificar. El directorio `examples` también contiene otros ejemplos (en un archivo denominado `parm_apps`) que se pueden copiar en el archivo `parm` y modificarlos si es preciso.

File

El parámetro `file` especifica los archivos de programa que pertenecen a una aplicación. Se incluyen todas las ejecuciones interactivas o en segundo plano de estos programas. Se aplica a la última instrucción `application` enviada. Se genera un error si no se encuentra ninguna instrucción `application`.

El nombre `file name` puede ser uno de los siguientes:

- Un único archivo de programa UNIX como `vi`.
- Un grupo de archivos de programa UNIX (indicados con un carácter comodín), como `xyz*`. En este caso, se incluye cualquier nombre de programa que comience con las letras `xyz`. Una especificación de archivo con caracteres comodines cuenta como una única especificación en relación con el máximo permitido.

El nombre del parámetro `file` está limitado a 15 caracteres. Se puede utilizar de manera opcional un signo igual (=) entre el parámetro `file` y el nombre de archivo.

Se pueden especificar varios nombres de archivo en la misma línea de parámetros (separados por comas) o en instrucciones de archivos separadas. Los nombres de archivos no pueden calificarse por un nombre de ruta. Las especificaciones de archivo se comparan con la métrica `PROC_PROC_NAME` específica, que se establece en un valor `argv[0]` del proceso (por lo general su nombre básico). Por ejemplo:

```
application = prog_dev
file = vi,vim,gvim,make,gmake,lint*,cc*,gcc,ccom*,cfront
file = cpp*,CC,cpass*,c++*
file = xdb*,adb,pxdb*,dbx,xlC,ld,as,gprof,lex,yacc,are,nm,gencat
file = javac,java,jre,aCC,ctcom*,awk,gawk
```

```
application Mail
file = sendmail,mail*,*mail,elm,xmh
```

Si no se especifica ningún parámetro `file`, se califican todos los programas que cumplen los otros parámetros.



El asterisco (*) es el único carácter comodín admitido por los calificadores de aplicación del archivo `parm` excepto para el calificador `cmd` (ver a continuación).

argv1

El parámetro `argv1` especifica los procesos seleccionados para la aplicación por el valor de la métrica `PROC_PROC_ARGV1`. Por lo general es el primer argumento de la línea de comandos, excepto cuando `javaarg=true`, si es el nombre `class` o `jar` de los procesos java. Este parámetro utiliza el mismo patrón de sintaxis coincidente utilizado por los parámetros de `parm`, como `file=` y `user=`. Cada criterio de selección puede llevar asteriscos como carácter comodín y se puede tener más de una selección en una línea separada por comas.

Por ejemplo, la siguiente definición de aplicación agrupa todos los procesos cuyo primer argumento de la línea de comandos es `-title`, `-fn` o `-display`:

```
application = xapps
argv1 = -title,-fn,-display
```

La siguiente definición de aplicación agrupa una aplicación java específica (si `javaarg=true`):

```
application = JavaCollector
argv1 = com.*Collector
```

A continuación se muestra cómo el parámetro `argv1` se puede combinar con el parámetro `file`:

```
application = sun-java
file = java
argv1 = com.sun*
```

cmd

El parámetro `cmd` especifica los procesos que se van a incluir en una aplicación por sus cadenas de comandos, que consta del programa ejecutado y sus argumentos (parámetros). A diferencia de otros parámetros de selección, este parámetro permite un uso extensivo de caracteres comodín además del uso del carácter de asterisco.

De la misma forma que en las expresiones regulares, se permite la coincidencia de patrón extensiva. Para una descripción completa de los criterios de patrón, consulte la página del manual de UNIX de `fnmatch`. A diferencia de otros parámetros, sólo se puede tener una selección por línea; sin embargo, puede tener varias líneas.

A continuación se muestra el uso del parámetro `cmd`:

```
application = newbie
cmd = *java *[Hh]ello[Ww]orld*
```

User

El parámetro `user` especifica qué usuarios (nombres de inicio de sesión) pertenecen a la aplicación. El formato es el siguiente:

```
application <nombre_de_aplicación>
file <nombre_de_archivo>
user [<nombre_de_dominio>]\<nombre_de_usuario>
```

El nombre de dominio del parámetro `user` es opcional. Hay que indicar el nombre de dominio para especificar los nombres de usuario de un sistema que no sea local.

Por ejemplo:

```
application test_app
file test
user TestDomain\TestUser
```

Si se especifica el nombre de usuario sin el nombre de dominio en el parámetro `user`, se considerará que los nombres de usuario son nombres de usuario del sistema local.

Por ejemplo:

```
application Prog_Dev
file vi,xb,abb,ld,lint
```

```
user ted,rebecca,test*
```

Sólo se puede usar el asterisco comodín (*) para asegurar que los nombres de usuario con una cadena similar de caracteres con prefijo antes del asterisco (*) y con sufijo después del asterisco (*) pertenecen a la aplicación. Si no especifica un parámetro `user`, se califican todos los programas que cumplen los otros parámetros.

El nombre del parámetro `user` está limitado a 15 caracteres.

Group

El parámetro `group` especifica qué nombres de grupos de usuario pertenecen a una aplicación.

Por ejemplo:

```
application Prog_Dev_Group2
file vi,xb,abb,ld,lint
user ted,rebecca,test*
group lab, test
```

Si no especifica un parámetro `group`, se califican todos los programas que cumplen los otros parámetros.

El nombre del parámetro `group` está limitado a 15 caracteres.

Or

El parámetro `or` se utiliza para permitir que más de una definición de aplicación se aplique a la misma aplicación. Dentro de una única definición de aplicación, el proceso debe coincidir al menos con una de las categorías de parámetros. Los parámetros separados por el parámetro `or` se consideran definiciones independientes. Si un proceso coincide con las condiciones de cualquier definición, en este caso pertenecerá a la aplicación.

Por ejemplo:

```
application = Prog_Dev_Group2
user julie
or
user mark
file vi, store, dmp
```

Lo anterior define la aplicación (`Prog_Dev_Group2`) que consta de un programa ejecutado por `user julie` más otros programas (`vi, store, dmp`) ejecutados por `user mark`.

Priority

Se pueden restringir los procesos de una aplicación a aquellos que pertenecen a un rango determinado especificando valores en el parámetro `priority`.

Por ejemplo:

```
application = swapping
priority 128-131
```

En función de su prioridad, los procesos se clasifican entre -511 y 255, dependiendo de la plataforma de Componente Performance Collection que se esté ejecutando. La prioridad puede modificarse durante todo el ciclo de vida de un proceso. El programador ajusta la prioridad de los procesos con recursos compartidos del tiempo. También se pueden cambiar las prioridades mediante programación o durante la ejecución.



El archivo `parm` se procesa en el orden introducido y la primera coincidencia del calificador definirá la aplicación a la que pertenece un proceso particular. Por consiguiente, es habitual disponer de las definiciones de aplicación más específicas antes de las más generales.

Ejemplos de definición de aplicación

En los ejemplos siguientes se muestran las definiciones de aplicación.

```
application firstthreesvrs
cmd = *appserver* *-option[123]*
application oursvrs
cmd = *appserver*
user = xyz,abc

application othersvrs
cmd = *appserver*
cmd = *appsvr*
or
argv1 = -xyz
```

A continuación se muestra un ejemplo de cómo se podrían registrar varios programas con el archivo `parm` anterior.

Cadena de comandos	Inicio de sesión de usuario	Aplicación
<code>/opt/local/bin/appserver -xyz -option1</code>	xyz	firstthreesvrs
<code>./appserver -option5</code>	root	othersvrs
<code>./appserver -xyz -option2 -abc</code>	root	firstthreesvrs
<code>./appsvr -xyz -option2 -abc</code>	xyz	othersvrs
<code>./appclient -abc</code>	root	other
<code>./appserver -mno -option4</code>	xyz	oursvrs
<code>appserver -option3 -jkl</code>	xyz	firstthreesvrs
<code>/tmp/bleh -xyz -option1</code>	xyz	othersvrs

Configuración de intervalos de registro de datos

Los intervalos de recopilación predeterminados utilizados por `scope` son 60 segundos para los datos de proceso y 300 segundos para las clases globales y para otras clases de datos. Se pueden invalidar estos valores con el parámetro `collectioninterval` del archivo `parm`. Los valores deben cumplir las condiciones siguientes:

- Se pueden configurar los intervalos de recopilación para los datos de proceso entre 5 a 60 segundos en pasos de 5 segundos. Los intervalos de recopilación para los datos de proceso deben ser un múltiplo del intervalo de `subproc` (consulte [subprointerval](#)), que se dividirá a partes iguales entre el intervalo de recopilación global.

- Se puede configurar el intervalo de recopilación para los datos globales en uno de los valores siguientes: 15, 30, 60 y 300 segundos. El intervalo de recopilación global debe ser mayor o igual que el intervalo del proceso, y un múltiplo del intervalo de recopilación del proceso. El intervalo de recopilación global se aplica a las métricas globales y a todas las clases de métrica que no son de proceso, como filesystem y application.

Ejemplo 1:

```
collectioninterval process=15, global=30
subprocinterval = 5
```

En este ejemplo, el intervalo de recopilación para los datos de proceso está establecido en 15 segundos, para las clases globales y otras clases de datos a 30 segundos y para subprocinterval en 5 segundos:

El intervalo de recopilación para los datos de proceso

- es un valor válido (este valor es un múltiplo de 5)
- es un múltiplo de subprocinterval, 5 segundos ($15\%5 = 0$)
- se divide en partes iguales en el intervalo de recopilación para datos globales, 30 segundos ($30\%15 = 0$)

El intervalo de recopilación para los datos globales

- es un valor válido (este valor puede ser 15, 30, 60 o 300)
- es un múltiplo del intervalo de proceso, 15 segundos ($15\%30 = 0$)

Por consiguiente estos valores de los intervalos de la recopilación son válidos.

Ejemplo 2:

```
collectioninterval process=15, global=30
subprocinterval = 10
```

En este ejemplo, el intervalo de recopilación para los datos de proceso está establecido en 15 segundos, para las clases globales y otras clases de datos a 30 segundos y para subprocinterval en 10 segundos:

El intervalo de recopilación para los datos de proceso

- es un valor válido (este valor es un múltiplo de 5)
- *NO* es un múltiplo de subprocinterval, 10 segundos ($15\%10 \neq 0$)
- se divide en partes iguales en el intervalo de recopilación global, 30 segundos ($30\%15 = 0$)

El intervalo de recopilación para los datos globales es de 30 segundos

- es un valor válido (este valor puede ser 15, 30, 60 o 300)
- es un múltiplo del intervalo de proceso, 15 segundos ($15\%30 = 0$)

Por consiguiente estos valores de los intervalos de la recopilación no son válidos.



Para Componente Performance Collection en VMware ESX Server, el intervalo de registro para los datos globales y todas las demás clases de datos que no sean datos de proceso se pueden configurar a 60 o 300 segundos.

Normalización de métricas de CPU en sistemas habilitados para hipersubprocesamientos/subprocesamientos múltiples simultáneos

En un sistema donde está habilitado hipersubprocesamientos/subprocesamientos múltiples simultáneos (HT/SMT), la CPU física admite dos o más subprocesos de hardware. Como resultado, varios procesos de software o subprocesos pueden ejecutarse simultáneamente en los subprocesos de hardware. En un sistema con un procesador multinúcleo, varios subprocesos pueden ejecutarse simultáneamente en núcleos individuales.

Performance Collection Component proporciona varias métricas relacionadas con CPU, que ayudan a analizar y comprender el uso de la CPU del sistema monitorizado. De manera predeterminada, en todos los sistemas habilitados para HT/SMT, Performance Collection Component calcula los valores de todas las métricas relacionadas con CPU normalizando los datos recopilados en el número de subprocesos disponibles en el sistema monitorizado. Cuando un único subproceso utiliza completamente todo el núcleo de la CPU, los valores calculados usando la normalización basada en subprocesos no siempre representa la imagen verdadera de la utilización de la CPU.

Esta versión de HP Operations Agent introduce un nuevo parámetro de configuración, `ignore_mt`, que permite configurar Componente Performance Collection para que registre los datos relacionados con la CPU que se han calculado usando la normalización basada en núcleo. Los valores de las métricas que se calculan con la normalización basada en el núcleo presentan un estado más preciso del uso de la CPU y, por tanto, ayudan a tomar decisiones más eficaces cuando se analice el rendimiento del sistema.

Registro de métricas calculadas con la normalización basada en núcleo

En HP-UX, se puede configurar Performance Collection Component para registrar todas las métricas relacionadas con la CPU con normalización basada en núcleo. En otras plataformas, se puede configurar Performance Collection Component para calcular las métricas relacionadas con la CPU de la clase GLOBAL con la normalización basada en núcleo antes del registro.

Para configurar Performance Collection Component con objeto de que use la normalización basada en núcleo para métricas relacionadas con la CPU, siga estos pasos:

En HP-UX

- 1 Inicie una sesión en el sistema con los privilegios raíz.
- 2 Configure el archivo `parm` en función de sus requisitos. No establezca el indicador `ignore_mt` del archivo `parm`.
 - ▶ El valor del indicador `ignore_mt` del archivo `parm` en HP-UX no tiene efecto en la operación de Performance Collection Component.
- 3 Defina las reglas de alarma si es preciso.
- 4 Ejecute el comando siguiente:
`midaemon -ignore_mt`

- 5 Inicie HP Operations Agent usando el comando siguiente:

```
opcagt -start
```

Performance Collection Component comienza el registro de todas las métricas relacionadas con la CPU (para todas las clases) usando la normalización basada en núcleo.

En otras plataformas

- 1 Inicie sesión en el sistema con privilegios raíz o administrativos.
- 2 Configure el archivo parm en función de sus requisitos. Establezca el indicador `ignore_mt` del archivo parm en true.
- 3 Defina las reglas de alarma si es preciso.
- 4 Inicie HP Operations Agent usando el comando siguiente:

En Windows

```
%ovinstalldir%bin\opcagt -start
```

En HP-UX, Linux o Solaris

```
/opt/OV/bin/opcagt -start
```

En AIX

```
/usr/lpp/OV/bin/opcagt -start
```

Performance Collection Component comienza el registro de todas las métricas relacionadas con la CPU para la clase GLOBAL usando la normalización basada en núcleo.

Detención y reinicio de la recopilación de datos

El recopilador `scope` y los otros procesos asociados están diseñados para ejecutarse continuamente. La única vez que se deberían detener es cuando ocurre una de las posibilidades siguientes:

- Está actualizando el software de Componente Performance Collection a una nueva versión.
- Está agregando o eliminando transacciones en el archivo de configuración de transacciones, `ttt.conf`. (Para obtener más información, consulte [Descripción del seguimiento de transacciones](#) en la página 347.)
- Está modificando rangos de distribución u objetivos de nivel de servicio (SLO) en el archivo de configuración de transacciones, `ttt.conf`. (Para obtener más información, consulte [Descripción del seguimiento de transacciones](#) en la página 347.)
- Está modificando el archivo `parm` y desea que los cambios surtan efecto. Los cambios realizados en el archivo `parm` surten efecto sólo cuando se inicia `scope`.
- Está utilizando el comando `resize` del programa `utility` para cambiar el tamaño del archivo de registro de Componente Performance Collection.
- Está apagando el sistema.
- Está agregando hardware o modificando los cambios de configuración. Los cambios realizados surtirán efecto sólo cuando se inicie `scope`.

Detención de la recopilación de datos

La opción `stop` de la secuencia de comandos `ovpa` y `mwa` asegura que no se pierde ningún dato cuando se detienen `scope` y otros procesos de Componente Performance Collection.

Para detener manualmente la recopilación de datos, escriba el comando siguiente:

- *En Windows:*
`%ovinstalldir%bin\ovpacmd -stop`
- *En HP-UX, Linux y Solaris:*
`/opt/perf/bin/ovpa -stop`
- *En AIX:*
`/usr/lpp/perf/bin/ovpa -stop`



`scope` no registra los datos de NFS, aunque estos datos se pueden ver con HP GlancePlus en el sistema de archivos local.

Reinicio de la recopilación de datos

Hay diferentes opciones para reiniciar la recopilación de datos después de detener los procesos de Componente Performance Collection o modificar los archivos de configuración y si se desea que estos cambios surtan efecto.

Para iniciar `scope` y los demás procesos de Componente Performance Collection una vez apagado el sistema, o después de haberlos detenido, se utiliza `<InstallDir>/ovpa start` si se está utilizando `coda`. En este caso, `InstallDir` es el directorio en donde se ha instalado Componente Performance Collection.

Para reiniciar `scope` y los demás procesos mientras se están ejecutando, se utiliza `<InstallDir>/ovpa restart` si se está utilizando `coda`. En este caso, `InstallDir` es el directorio en donde se ha instalado Componente Performance Collection. Esto detiene los procesos actualmente en ejecución y los vuelve a iniciar.

Cuando se reinicia `scope`, Componente Performance Collection continúa usando los mismos archivos de registro (`logglob`, `logappl`, `logproc`, `logdev`, `logtran`, `logls` y `logindx`) utilizados antes de detener el programa. Se anexan nuevos registros al final de los archivos existentes. Si se desea recopilar datos en un nuevo conjunto de archivos y no conservan ninguna información histórica, hay que cambiar el nombre o almacenarlos y eliminar *todos* los archivos de registros de `scope` antes de reiniciar porque los datos se sincronizan entre los archivos.



La entrada `SEM_KEY_PATH` del archivo de configuración `ttd.conf` se usa para generar claves IPC para los semáforos usados en los procesos `ttd` y `midaemon` en las plataformas UNIX. El valor predeterminado usado es `/var/opt/perf/datafiles`.

Se puede cambiar el valor de `SEM_KEY_PATH` si `midaemon` o `ttd` no responden debido a colisiones entre los identificadores de semáforos.

Horario de verano

Durante el horario de verano, la hora del sistema se atrasa una hora en las zonas horarias correspondientes. En este punto, la recopilación de datos se detiene durante una hora hasta que la hora del sistema se sincronice con la marca de hora del último registro.

Cuando se desactiva el horario de verano, la hora del sistema adelanta una hora y, por tanto, la marca de hora del siguiente registro adelanta una hora. Este cambio horario produce una brecha de una hora después del último registro inscrito, aunque la recopilación de datos no se detiene.

Cambio manual de la hora del sistema

Cuando la hora del sistema se retrasa manualmente, la recopilación de datos y comandos como **perfstat** no funcionan. Estas utilidades se detienen cuando la hora del sistema se retrasa. Para continuar registrando datos y para obtener respuestas de todos los comandos, ejecute los pasos siguientes:

- 1 Ejecute el comando siguiente:

```
ovc -stop coda
```

- 2 Realice una copia de seguridad de los archivos `coda.*` del directorio `<dir_de_datos>\datafiles\` y suprámalos.

- 3 Ejecute el comando siguiente:

```
ovc -start coda
```

Administración eficaz de la recopilación de datos

El análisis eficaz de rendimiento depende de lo fácil que sea acceder a los datos de rendimiento que se recopilen. En esta sección se describen estrategias eficaces para actividades como la administración de archivos de registro, el almacenamiento de datos y el análisis del sistema para que el proceso de recopilación de datos sea más sencillo, más eficaz y más útil.

Control del espacio en disco usado por los archivos de registro

Componente Performance Collection incluye la administración automática de los archivos de registro que crea. Se puede configurar este proceso automático o utilizar procesos manuales alternativos con fines concretos. El proceso de administración automática del archivo de registro funciona de la forma siguiente:

- Cada archivo de registro tiene un tamaño máximo configurado. Los tamaños máximos predeterminados se incluyen cuando se instala Componente Performance Collection por primera vez. Sin embargo, se pueden volver a configurar estos valores.
- Cuando cada archivo de registro alcanza su tamaño máximo, el recopilador de datos `scope` realiza una "reversión" en `mainttime`. Durante esta reversión, se suprimen el 25 por ciento de los datos más antiguos del archivo de registro para agregar nuevos datos.

El mantenimiento automático del archivo de registro es similar, pero no idéntico, para los datos recopilados por `scope` y por el proceso de registro de DSI. Para obtener más información en el mantenimiento del archivo de registro de DSI, consulte [Introducción general de Integración de Orígenes de Datos \(DSI\)](#) en la página 263.

Configuración de mainttime

Por lo general, `scope` sólo realizará reversiones de archivos de registro en un período de tiempo específico cada día. De esta forma se garantiza que la operación se va a realizar fuera de horas pico y que no impacta en el uso normal del sistema. La hora en que se examinan los archivos de registro para la reversión se establecen en el parámetro `mainttime` del archivo `parm`.

Establecimiento del tamaño máximo del archivo de registro

La elección de un tamaño máximo del archivo de registro debería estar equilibrada entre cuánto espacio en disco se utiliza y cuántos datos históricos están disponibles para el análisis inmediato. Unos archivos de registro con menor tamaño ahorran espacio en disco, pero limitan el tiempo en que se pueden representar gráficamente por herramientas como Performance Manager. A continuación se describen algunas formas de volver a configurar el tamaño de los archivos de registro de `scope`.

`scope` registra distintos tipos de datos en sus propios archivos de registro. De esta forma se puede elegir cuánto espacio en disco se desea dedicar a cada tipo de manera independiente. Por ejemplo, los datos globales son bastante compactos pero con frecuencia se quiere retroceder en el tiempo y representar gráficamente los datos durante un mes. Este sistema permite disponer de una buena base estadística para los ejercicios de planificación de capacidad y de representación de tendencias.

Los datos de proceso pueden consumir más espacio en disco que los datos globales porque es posible disponer de muchos procesos interesantes cada minuto. Además, la relación tiempo-valor de los datos de proceso no es tan elevada como en los datos globales. Puede ser muy importante conocer los detalles sobre qué procesos se están ejecutando hoy y cuáles lo hicieron ayer. Ocasionalmente es posible que haya que saber qué procesos se ejecutaron la última semana. Sin embargo, es muy poco probable que resulte útil saber exactamente qué procesos se ejecutaron el último mes.

Un usuario típico podría decidir mantener los datos siguientes en línea:

- Tres meses de datos globales con fines de análisis de tendencias
- Un mes de datos de proceso para la solución de problemas
- Tres meses de datos de aplicación con fines de análisis de tendencias y equilibrio de carga
- Dos meses de datos de dispositivo para el equilibrio de carga del disco

Se puede editar el archivo `parm` para establecer los parámetros `size` para cada archivo de registro diferente. Los tamaños se especifican en megabytes. Por ejemplo:

```
SIZE GLOBAL=10.0 PROCESS=30.0 APPLICATION=20.0 DEVICE=5.0
```

El número de megabytes requeridos para almacenar un número determinado de días puede variar por tipo de datos, configuración del sistema y actividad del sistema. La mejor forma de determinar el tamaño de los archivos de registro en el sistema es recopilar los datos durante una semana aproximadamente y, después, utilizar el comando `resize` del programa `utility` para cambiar el tamaño del archivo de registro. El comando `resize` examina los archivos de registro y determina cuántos datos se registran cada día. Después los convierte de días a megabytes. Esta función también actualiza el archivo `parm`.

Administración de los procesos de cambio de tamaño

No se requieren actividades adicionales una vez configurado el mantenimiento automático del archivo de registro. Cuando los archivos de registro alcanzan su tamaño máximo configurado, `scope` cambiará automáticamente su tamaño.

`scope` reierte los archivos de registro al valor de `mainttime` especificado en el archivo `parm`. Si se edita el archivo `parm` y reinicia `scope`, los archivos de registro no adoptarán los nuevos tamaños hasta que se alcance el valor de `mainttime`. Es importante que `scope` se esté ejecutando en la hora especificada en `mainttime` o los archivos de registro nunca se podrán revertir.

Los archivos de registro pueden exceder su tamaño máximo configurado entre las horas de mantenimiento sin causar una reversión inmediata.

Un archivo de registro nunca cambiará su tamaño para almacenar menos datos a los correspondientes a un día completo. Esto significa que el archivo de registro podrá aumentar para almacenar al menos los datos correspondientes a un día antes de revertirse. Esto no suele constituir un problema, pero si se establecen los parámetros del archivo `parm` para recopilar un gran volumen de datos de proceso o de aplicación o se establecen en un tamaño demasiado pequeño, el resultado de un archivo de registro podría exceder de manera significativa su tamaño máximo configurado antes de su reversión.

`scope` comprueba el espacio en disco disponible en el sistema de archivos donde residen los archivos de registro, periódicamente a intervalos especificado en el archivo `parm` para la recopilación de datos globales. Si el espacio en disco disponible es inferior a un megabyte, `scope` toma medidas para asegurar que no usa más espacio disponible de la manera siguiente:

- Realiza inmediatamente el mantenimiento del archivo de registro sin esperar el tiempo de mantenimiento regular del archivo de registro. Si cualquier archivo de registro excede su tamaño máximo (y tiene más datos correspondientes a un día), se revertirá.
- Si, después del mantenimiento del archivo de registro, el espacio en disco disponible sigue siendo inferior a un megabyte, `scope` escribe un mensaje de error apropiado en el archivo `status.scope` e interrumpe la recopilación de datos.

Almacenamiento de datos

La administración automática de los archivos de registro mantiene los datos del archivo de registro más recientes disponibles para su análisis. Los datos de los archivos de registro sin procesar se almacenan. Los datos del proceso y los datos globales se registran periódicamente a intervalos especificados en el archivo `parm`. Para obtener más información, consulte [Configuración de intervalos de registro de datos](#) en la página 37. Para dejar espacio a nuevos datos, los datos más antiguos se suprimen cuando los archivos de registro llegan a su tamaño máximo. Si se desea mantener los datos de los archivos de registro durante períodos de tiempo superiores, se debería establecer un proceso de almacenamiento de datos. El proceso exacto que se elija depende de las necesidades del usuario. A continuación se indican algunas posibilidades:

- Cambiar el tamaño de los archivos de registro sin procesar para que sean más grandes y dejar que el mantenimiento automático del archivo de registro haga el resto. Este es el método de almacenamiento más sencillo, pero puede consumir grandes cantidades de espacio en disco después de varios meses.
- Extraer los datos de los archivos de registro sin procesar en los archivos de almacenamiento extraídos antes de que se supriman de los archivos de registro sin procesar. Establecer un procedimiento para copiar los archivos de registro en un almacenamiento a largo plazo, como una cinta, hasta que se necesiten.

- Extraer sólo un subconjunto de los archivos de registro sin procesar en los archivos de almacenamiento extraídos. Por ejemplo, es posible que no se deseen almacenar datos de proceso debido a su elevado volumen y a su baja relación tiempo-valor.
- Se pueden usar algunas de las técnicas anteriores combinadas.

Recomendamos los procedimientos siguientes para el almacenamiento de datos:

- Cambiar el tamaño de los archivos de registro sin procesar para acomodar la cantidad de datos detallados que se desean mantener en línea.
- Una vez a la semana, copiar los datos sin procesar detallados en archivos que se pueden desplazar a un almacenamiento sin conexión.

Administración de los procesos de almacenamiento

Se puede cambiar el tamaño de los archivos de registro sin procesar, tal como se describió en la sección anterior. Se eligen tamaños de los archivos de registro para que almacenen al menos datos correspondientes a dos semanas (asumiendo que el procesamiento de almacenamiento se realizará sólo una vez a la semana).

Una vez a la semana, se programa un proceso que ejecuta el programa `extract`. En el ejemplo siguiente se muestra una secuencia de comandos que ejecuta el procesamiento semanal:

```
#extract -gapdt -xm
```

Cada semana durante el mes se anexarán los datos a los datos de la semana anterior. Cuando comienza un nuevo mes, `extract` crea un nuevo archivo de registro de almacenamiento y divide los datos de esa semana en los archivos de registro de almacenamiento mensuales apropiados. Los archivos de registro se denominan `rxmo`, seguidos por cuatro dígitos para el año y dos dígitos más para el mes. (Por ejemplo, los datos correspondientes a diciembre de 1999 podrían estar disponibles en un archivo denominado `rxmo199912`.)

Al comienzo de cada mes, se completa el archivo de registro del mes anterior y se inicia un nuevo archivo de registro. Por consiguiente, siempre que esté presente más de un archivo de registro `rxmo`, es seguro copiar todos excepto el último en un almacenamiento sin conexión hasta que sea necesario. Cuando sea preciso acceder a los datos almacenados, se restaura el archivo de almacenamiento deseado y se accede a él mediante los programas `extract` o `utility`.

Dependiendo de la configuración del sistema y de los niveles de actividad, la cantidad de espacio en disco acumulada en un mes puede ser grande. Si éste fuera el caso, se puede desglosar el archivo de almacenamiento detallado en archivos más pequeños sustituyendo el comando semanal `-xw` en lugar de `-xm`, tal como se muestra en el ejemplo.

Otra alternativa es optar por no almacenar los datos de proceso detallados.

La extracción detallada que se trató en el ejemplo anterior conserva todos los datos de rendimiento recopilados. Si alguna vez hay que investigar una situación en profundidad, estos archivos se pueden restaurar en el disco y analizarse.



El programa `extract` se utiliza para combinar datos procedentes de varios archivos extraídos o crear un subconjunto de los datos para un transporte y análisis más sencillos.

Por ejemplo, se pueden combinar datos de varios archivos anuales extraídos para realizar análisis de tendencias de varios años.



No se admite el traslado de archivos de registro creados en un nodo de HP Operations Agent a un sistema que utilice una versión anterior de HP Performance Agent.

3 Cómo trabajar con HP Operations Agent

Después de configurar el mecanismo de recopilación de datos, si se desea usar el agente con HPOM, se pueden usar distintos componentes de Operations Monitoring Component mediante la implementación de directivas HPOM en el nodo. Por ejemplo, si se implementa una directiva de umbral de medida, el componente del agente de monitorización comienza a funcionar. Aunque se proporcione la mayoría de los detalles de monitorización en las directivas de HPOM, es posible que algunos componentes todavía requieran que se realice una configuración adicional en el nodo.

Configuración del agente de monitorización

Se puede iniciar y configurar el agente de monitorización para monitorizar diferentes orígenes. Si se implementa una directiva de umbral de medida en un nodo, el componente del agente de monitorización comienza a funcionar.

En función de la especificación de las directivas, el agente comienza a monitorizar objetos desde los siguientes tipos de orígenes:

- **Externo:** Programa externo que puede enviar valores numéricos al agente.
- **Componente de rendimiento incrustado:** Los datos disponibles en el almacén de datos del agente.
- **MIB:** Entradas de la Base de información de administración (MIB).
- **Administración de rendimiento en tiempo real:** Registros y alertas de rendimiento de Windows.
- **Programa:** Un programa externo iniciado por HPOM y envía valores numéricos al agente.
- **WMI:** La base de datos WMI.

Para utilizar las directivas HPOM con objeto de monitorizar los objetos de los orígenes anteriores, consulte los temas siguientes:

- *En HPOM para Windows:* La sección *Event Policy Editors* de la ayuda *HPOM for Windows Online Help*.
- *En HPOM en UNIX/Linux:* La sección *Implementing Message Policies* de la guía *HPOM for UNIX 9,10 Concepts Guide*.

Configuración del agente para monitorizar objetos MIB

Después de implementar las directivas de umbral de medición (con el tipo de origen establecido en MIB) en el nodo, el agente de monitorización comienza a consultar los objetos MIB a los que se puede acceder con la cadena de comunidad `public`. Si se desea configurar el agente de monitorización para que utilice una cadena de comunidad que no sea la predeterminada, siga estos pasos:

- 1 Inicie sesión en el nodo con privilegios raíz o administrativos.
- 2 Vaya al símbolo del sistema (shell).
- 3 Vaya al directorio siguiente:

Windows:

`%ovinstalldir%bin`

HP-UX, Solaris y Linux:

`/opt/OV/bin`

AIX:

`/usr/lpp/OV/bin`

- 4 Ejecute el comando siguiente:

- Para utilizar una cadena de comunidad que no sea la predeterminada:

```
ovconfchg -ns eaagt SNMP_COMMUNITY <cadena_de_comunidad>
```

En este ejemplo, `<cadena_de_comunidad>` es la cadena de comunidad no predeterminada que ha elegido.

- Para utilizar cadenas de comunidad diferentes:

```
ovconfchg -ns eaagt SNMP_COMMUNITY_LIST <lista_de_cadenas_comunidad>
```

En este ejemplo, `<lista_de_cadenas_comunidad>` es la lista separada por comas de cadenas de comunidad que ha elegido. HP Operations agent procesa la lista de cadenas de comunidad en el orden especificado con el comando anterior.

Por ejemplo:

```
ovconfchg -ns eaagt SNMP_COMMUNITY_LIST "C1,C2,C3"
```

HP Operations agent intenta primero establecer una sesión SNMP con los nodos y ejecutar una operación `SNMP Get` para los OID usando la cadena de comunidad `C1`. Si la operación no es correcta, HP Operations agent ejecuta la misma operación con la cadena de comunidad `C2`, y así sucesivamente.



Si HP Operations agent no puede utilizar todas las cadenas de comunidad especificadas con `SNMP_COMMUNITY_LIST`, intenta usar la cadena de comunidad especificada con `SNMP_COMMUNITY`. Si el agente no puede obtener los datos con todas las cadenas de comunidad especificadas, comienza a usar la cadena de comunidad `public` predeterminada.

Monitorización de OID secundarios

Al crear una directiva de umbral de medida MIB, hay que establecer el campo de MIB ID para el OID del objeto MIB que se desea monitorizar. Si se desea monitorizar los OID secundarios del objeto, hay que escribir el OID en el campo MIB ID con el formato siguiente:

```
<OID_principal>.*
```


Por ejemplo, si se desea monitorizar todos los OID secundarios del objeto `.1.3.6.1.2.1.2.2.1`, hay que especificar `.1.3.6.1.2.1.2.2.1.*` en el campo MIB ID.

También se puede especificar la posición del nombre o descripción de las instancias secundarias con el formato siguiente:

`<OID_principal>.*:<n>`

En este caso, `<n>` es la posición del nombre o descripción de las instancias secundarias.

Por ejemplo, si se desea monitorizar todos los OID secundarios del objeto `.1.3.6.1.2.1.2.2.1` que se describen por el segundo elemento en el subárbol MIB para `.1.3.6.1.2.1.2.2.1`, hay que especificar `.1.3.6.1.2.1.2.2.1.*:2` en el campo MIB ID.

Persistencia de un objeto monitorizado

Esta versión de HP Operations agent almacena periódicamente los valores de los objetos monitorizados y variables de sesión para conservar los valores en caso de una interrupción o error. Sin embargo, se puede configurar el agente para que no conserve los valores de los objetos monitorizados y variables de sesión.

Para configurar el agente con objeto de que no conserve los valores monitorizados, siga estos pasos:

- 1 Inicie sesión en el nodo con los privilegios necesarios.
- 2 En el símbolo del sistema, ejecute el siguiente comando:

```
ovconfchg -ns eaagt set OPC_MON_SAVE_STATE FALSE
```

De ahora en adelante, el agente detiene la conservación de los valores de los objetos monitorizados y variables de sesión.

Configuración del interceptor de eventos

De manera predeterminada, el interceptor de eventos puede recopilar capturas SNMP que se originan en las estaciones de administración remotas o en dispositivos habilitados para SNMP y, a continuación, generar eventos apropiados basados en la configuración. Se puede modificar el comportamiento predeterminado del interceptor de eventos mediante la configuración de las propiedades siguientes:

- `SNMP_TRAP_PORT`: El puerto predeterminado es **162**. Se puede modificar este valor en cualquier puerto disponible en el nodo de HP Operations agent.
- `SNMP_TRAP_FORWARD_DEST_LIST`: Con esta propiedad, se puede establecer la dirección de la estación de administración remota en donde se deseen reenviar todas las capturas SNMP disponibles. Se pueden especificar varios nombres de sistema (con detalles de puerto) separados por comas.
- `SNMP_TRAP_FORWARD_ENABLE`: De manera predeterminada, esta propiedad está establecida en **FALSE**. Al establecer esta propiedad en **TRUE**, se puede habilitar el interceptor de eventos para reenviar las capturas SNMP disponibles en el nodo de HP Operations agent a equipos remotos o estaciones de administración.

- `SNMP_TRAP_FORWARD_COMMUNITY`: Con esta propiedad, se puede especificar la cadena de comunidad de los equipos de origen de las capturas entrantes y el equipo de destino donde desee reenviar las capturas SNMP. Las cadenas de comunidad de los equipos de origen deben coincidir con las de los equipos de destino.
- `SNMP_TRAP_FORWARD_FILTER`: Con esta propiedad, se pueden filtrar las capturas SNMP disponibles por sus OID y reenviar sólo las capturas seleccionadas al equipo remoto. El mecanismo de filtro surte efecto con el carácter comodín (*). Por ejemplo, si se establece esta propiedad en `1.2.3.*.*.*`, el interceptor de eventos reenviará todas las capturas SNMP con los OID que comienzan por `1.2.3`. De manera predeterminada, todas las capturas disponibles se reenvían cuando se habilita el interceptor de eventos para que reenvíe capturas.



Si la cadena de comunidad de los equipos de origen no coinciden con la de los equipos de destino, la función de reenvío de la captura no se realiza correctamente.

Para modificar el comportamiento predeterminado del interceptor de eventos, siga estos pasos:

- 1 Inicie sesión en el nodo con los privilegios necesarios.
- 2 En el símbolo del sistema, ejecute los comandos siguientes:
 - Para modificar el número de puerto, ejecute el comando siguiente:


```
ovconfchg -ns eaagt set SNMP_TRAP_PORT <número_de_puerto>
```

Hay que especificar un valor entero para `<número_de_puerto>`.
Hay que asegurarse de que el `<número_de_puerto>` especificado está disponible para su uso.
 - Para habilitar el interceptor de eventos para reenviar capturas SNMP a equipos remotos, ejecute el comando siguiente:


```
ovconfchg -ns eaagt set SNMP_TRAP_FORWARD_ENABLE TRUE
```
 - Si se habilita el interceptor de eventos para que reenvíe capturas SNMP a un equipo remoto, ejecute el comando siguiente para especificar los detalles del equipo de destino:


```
ovconfchg -ns eaagt set SNMP_TRAP_FORWARD_DEST_LIST "<nombre_de_equipo>:<puerto>"
```

`<nombre_de_equipo>` es el nombre de dominio completo del equipo en donde se desea reenviar las capturas SNMP y el `<puerto>` es el puerto HTTPS del equipo. Si se desea especificar varios destinos, separe los detalles del equipo con comas.
 - Si desea reenviar sólo determinadas capturas SNMP disponibles en el nodo al equipo remoto, ejecute el comando siguiente:


```
ovconfchg -ns eaagt set SNMP_TRAP_FORWARD_FILTER "<Filtro OID>"
```

`<Filtro OID>` es un OID con los caracteres de comodín. El interceptor de eventos filtra las capturas que coinciden con el OID especificado (con los caracteres comodín) desde las capturas disponibles y después las reenvía al equipo de destino.

Configuración del componente RTMA

El componente Real-Time Metric Access (RTMA) ofrece al usuario acceso en tiempo real a la métrica de rendimiento del sistema, tanto de manera local como remota. El proceso `perfd`, que forma parte del componente RTMA, comienza a ejecutarse en el nodo con la configuración predeterminada

después de instalar HP Operations agent. Se pueden modificar los ajustes de la configuración del proceso `perfd` en el archivo `perfd.ini`, que está disponible en el siguiente directorio del nodo:

- En Windows: `%ovdatadir%`
- En UNIX (y Linux): `/var/opt/perf`

Parámetro	Descripción	Valor predeterminado
<code>interval</code>	La frecuencia de la recopilación de datos en segundos. Este valor debe ser un múltiplo o factor de 60.	10
<code>port</code>	El puerto usado por <code>perfd</code> .	5227
<code>depth</code>	El período de tiempo en el que se conservan los valores de métricas globales en la caché de <code>perfd</code> . Estos datos se utilizan para el resumen de datos.	30
<code>maxrps</code>	El número máximo de solicitudes de sesión por segundo aceptadas por <code>perfd</code> . Si el número de solicitudes excede el límite, <code>perfd</code> hace una pausa de un segundo y, a continuación, registra los detalles de este evento en el archivo de registro. El archivo de registro, <code>status-perfd.<puerto></code> , está ubicado en el directorio siguiente del nodo: <ul style="list-style-type: none">• <i>En Windows:</i> <code>%ovdatadir%</code>• <i>En UNIX (y Linux):</i> <code>/var/opt/perf</code>	20
<code>maxtpc</code>	El número máximo de sesiones por sistema de cliente aceptadas por <code>perfd</code> . Cuando el número de disponible de sesiones alcanza este límite, si llega una solicitud adicional, <code>perfd</code> la deniega.	30

Parámetro	Descripción	Valor predeterminado
maxcps	El número máximo de solicitudes de sesión simultáneas aceptadas por perfd en un momento determinado. Si el número de solicitudes excede el límite, el servidor hará una pausa durante 3 segundos antes de establecer las sesiones.	2
lightweight	Si se establece en true, perfd detiene la recopilación de datos para los procesos, aplicación, operaciones NFS, sistemas lógicos y ARM. Además, no se recopilarán los datos HBA y LVM en HP-UX.	false
localonly	Si se establece en true, perfd sólo se puede configurar en el equipo local. Si se establece en true, perfd deniega todas las solicitudes de conexión excepto las que proceden del sistema host (localhost) a través de la interfaz de bucle invertido. Los detalles de las solicitudes de conexión denegadas se registran en el archivo de estado.	false
ipv4	Esta opción permite a perfd aceptar únicamente conexiones IPv4. De manera predeterminada, si perfd no puede crear un socket IPv6, cambia automáticamente al socket de IPv4. Esta opción se utiliza si se desea explícitamente deshabilitar cualquier comunicación IPv6.	false

Para cambiar la configuración predeterminada, siga estos pasos:

- 1 En el nodo, abra el archivo `perfd.ini` con un editor de texto.
- 2 Modifique la configuración.
- 3 Guarde el archivo.
- 4 Reinicie HP Operations agent para que los cambios surtan efecto.

Configuración del usuario del agente

- ▶ Aunque Operations Monitoring Component se puede ejecutar en un usuario no raíz o sin privilegios, Performance Collection Component siempre debe ejecutarse con un usuario raíz o administrativo. Si se configura HP Operations agent para que se ejecute con un usuario que no tenga los privilegios raíz o administrativos, no se podrá usar Performance Collection Component, y lo que es más importante, no se podrá recopilar ningún dato de rendimiento de sistema.

HP Operations agent, después de la instalación, comienza a ejecutarse con la cuenta de sistema local en los nodos de Windows y con la cuenta raíz en los nodos de UNIX/Linux. Sin embargo, se puede configurar el agente para que se ejecute con una cuenta no predeterminada.

El agente inicia comandos automáticos o iniciados por el operador en la cuenta del usuario en la que se está ejecutando el mismo agente. Sin embargo, se puede configurar el agente para que inicie los comandos en otra cuenta de usuario.

Cambio del usuario predeterminado en Windows

El usuario de agente que se desea usar en Windows debe cumplir los requisitos siguientes:

- Derechos de usuario para ejecutar el agente:
 - Iniciar sesión como un servicio
 - Administrar el registro de seguridad y auditoría
- Derechos de usuario para administrar el nodo:
 - Cerrar el sistema
De esta forma el agente cierra el sistema (por ejemplo, cuando un usuario inicia la herramienta de cierre en la consola).
 - Depurar programas
Permite al agente recopilar la información sobre los procesos y cerrar procesos (por ejemplo, cuando un usuario inicia los procesos de lista o la herramienta del proceso de cierre en la consola).
- Derechos de usuario para que el agente pueda iniciar comandos y herramientas como un usuario distinto al usuario de agente:
 - Actuar como parte del sistema operativo.
 - Ajustar cuotas de memoria para un proceso (también llamado Aumentar cuotas en algunas versiones de Windows)
 - Reemplazar un token de nivel de proceso.
- Permisos para las entradas del Registro:
 - HKEY_LOCAL_MACHINE/Software/Hewlett-Packard/OpenView
El usuario debe tener control completo para esta clave del Registro y todos los objetos secundarios.
 - HKEY_LOCAL_MACHINE/Software/Microsoft/WindowsNT/CurrentVersion/Perflib

El usuario debe tener permiso para leer esta clave del Registro para que el agente acceda a los datos de rendimiento.

Es posible que haya que asignar derechos adicionales para las tareas de administración que se deban realizar. Por ejemplo:

- Si se desea monitorizar un archivo de registro con una directiva, el usuario del agente debe tener permiso para leer dicho archivo de registro.
- Si se desea iniciar un programa mediante un comando automático, un comando iniciado por el operador, herramienta o tarea programada, el usuario del agente debe tener permiso para iniciar ese programa.

Además, hay que establecer el parámetro `OPC_PROC_ALWAYS_INTERACTIVE=NEVER` en el espacio de nombres `eaagt`. Este parámetro se puede configurar en los valores predeterminados de la instalación del agente o usando `ovconfchg` u `ovconfpar` en el símbolo del sistema. Después de establecer este parámetro, los procesos que inicia el agente no tendrán acceso al escritorio predeterminado. Esta configuración se aplica al preprocesamiento del encapsulador de archivos de registro y a las secuencias de comandos que llama el agente de monitorización.

- Algunos complementos inteligentes pueden requerir una configuración adicional o derechos de usuario cuando el agente se ejecuta en una cuenta de usuario que carece de derechos administrativos. Para obtener más información, consulte la documentación de los complementos inteligentes individuales.

Para cambiar el usuario de agente predeterminado en Windows, siga estos pasos:

- 1 *Opcional.* Cree un nuevo usuario con el que se ejecute el agente.
- 2 *Opcional.* Cree un nuevo grupo y agregue el usuario como miembro de este grupo.
- 3 En el nodo, abra el símbolo del sistema y escriba el siguiente comando:

```
cscript "%OvInstallDir%bin\ovswitchuser.vbs" -existinguser  
<dominio\usuario> -existinggroup <grupo> -passwd <contraseña>
```

En este ejemplo:

`<dominio\usuario>` es el dominio y el nombre de usuario.

`<grupo>` es el nombre del grupo al que pertenece el usuario, por ejemplo AgentGroup.

`<contraseña>` es la contraseña del usuario.

► El comando asigna los derechos de usuario requeridos para la funcionalidad básica del agente a nivel de grupo, no al del usuario individual. Por consiguiente, hay que prestar mucha atención cuando se selecciona el grupo que se va a usar. Se recomienda crear un nuevo grupo específicamente para el usuario de agente y agregar al usuario de agente como miembro.

- 4 Para reiniciar el agente, ejecute los siguientes comandos:

a **ovc -kill**

b **ovc -start**

Los procesos de agente y de servicio de control ahora se ejecutan como el usuario especificado.

Configure el usuario del agente para que inicie o pare servicios y procesos

Durante el transcurso de la operación, el agente puede necesitar iniciar o detener servicios y procesos (incluidos los propios procesos del agente). Deberá configurar el usuario, el cual ha sido configurado para ejecutar el agente, de manera que obtenga los privilegios para iniciar o parar servicios y procesos.

Si desea que el usuario del agente inicie o pare servicios en el nodo de Windows, siga los pasos siguientes:

- 1 Inicie una sesión en el nodo con privilegios administrativos.
- 2 Vaya al directorio `%ovinstalldir%\bin\xpl`.
- 3 Ejecute el comando siguiente:

```
ovsetscmpermissions.vbs -user <agente_de_usuario> -f
```

- 4 Reinicie el agente ejecutando los siguientes comandos:
 - **ovc -kill**
 - **ovc -start**

Cambio del usuario predeterminado en UNIX/Linux

De manera predeterminada, en los nodos con un sistema operativo UNIX o Linux el agente se ejecuta en la cuenta raíz. Sin embargo, se puede configurar el agente para que se ejecute en otra cuenta de usuario. Por ejemplo, se puede desear que el agente se ejecute en una cuenta con menos permisos que la cuenta raíz. Además, se puede desear que el agente se ejecute en una cuenta que tenga permiso para acceder a sistemas remotos en la red.

Hay que comprobar si la cuenta de usuario tiene los derechos apropiados para ejecutar el agente y administrar el nodo correctamente. Es posible que haya que asignar derechos adicionales para las tareas de administración que se deban realizar. Por ejemplo:

- Si se desea monitorizar un archivo de registro con una directiva, el usuario del agente debe tener permiso para leer dicho archivo de registro.
- Si se desea iniciar un programa mediante un comando automático, un comando iniciado por el operador, herramienta o tarea programada, el usuario del agente debe tener permiso para iniciar ese programa.
- Algunos complementos inteligentes pueden requerir una configuración adicional o derechos de usuario cuando el agente se ejecuta en un usuario alternativo. Para obtener más información, consulte la documentación de los complementos inteligentes individuales.

Para cambiar el usuario predeterminado en UNIX/Linux, siga estos pasos:

- 1 *Opcional.* Cree un nuevo usuario con el que se ejecute el agente.
- 2 *Opcional.* Cree un nuevo grupo y agregue después el usuario como miembro de este grupo.
- 3 En el nodo, inicie sesión como raíz y abra un símbolo del sistema.
- 4 Vaya al directorio siguiente:

En HP-UX, Linux o Solaris

```
/opt/OV/bin
```

En AIX

```
/usr/lpp/OV/bin
```

- 5 Detenga el agente ejecutando el siguiente comando:

```
ovc -kill
```

- 6 Cambie el usuario de agente ejecutando el siguiente comando:

```
ovswitchuser.sh -existinguser <usuario> -existinggroup <grupo>
```

En este ejemplo:

<usuario> es el nombre del usuario en el que se ejecuta el agente.

<grupo> es el nombre del grupo al que pertenece el usuario, por ejemplo AgentGroup. El comando da a este grupo control total de todos los archivos en el directorio de datos del agente y también control total de todos los paquetes instalados. Si se ha iniciado anteriormente el comando y se ha especificado un grupo diferente, el comando suprime el control de los archivos para el grupo anterior.

El indicador de ID del grupo se establece en los directorios de datos del agente. Este indicador significa que el grupo especificado también poseerá todos los nuevos archivos y subdirectorios de los directorios base del agente.



El comando asigna los derechos de usuario requeridos para la funcionalidad básica del agente a nivel de grupo, no al del usuario individual. Por consiguiente, hay que prestar mucha atención cuando se selecciona el grupo que se va a usar. Se recomienda crear un nuevo grupo específicamente para el usuario de agente y agregar al usuario de agente como miembro.

- 7 De manera predeterminada, HP Operations agent incluye agentes de comunicación que atienden conexiones entrantes desde los servidores de administración en el puerto 383. Sin embargo, en los nodos de UNIX y Linux, los usuarios que no son raíz no pueden abrir puertos en el rango 0 a 1023. Por tanto, hay que configurar el agente de comunicación en el nodo para que atienda en un puerto diferente (superior a 1023). También hay que configurar los servidores de administración que se conectan al nodo, para que sus conexiones salientes estén destinadas al puerto correcto.

Hay que configurar los puertos del agente de comunicación estableciendo el parámetro PORTS en el espacio de nombres `bbc.cb.ports`. Este parámetro se puede configurar de las siguientes formas:

- Configure los valores en los ajustes predeterminados de la instalación del agente. Esto se recomienda si se precisa configurar los puertos del agente de comunicación para un gran número de nodos. Hay que planificar y configurar los valores predeterminados de la instalación antes de crear o migrar los nodos.
- Se utiliza `ovconfchg` u `ovconfpar` en el símbolo del sistema.

El valor debe contener uno o más nombres de host o direcciones IP y tener el formato siguiente:

```
<host>:<puerto>[,<host>:<puerto>] ...
```

Por ejemplo, para configurar el puerto del agente de comunicación en 5000 en un nodo con el nombre de host `node1.emea.example.com`, utilice el comando siguiente en el mismo nodo y también en cualquier servidor de administración que abra conexiones a él.

```
ovconfchg -ns bbc.cb.ports -set PORTS  
node1.domain.example.com:5000
```

- 8 Para iniciar el agente, ejecute los siguientes comandos:

- a **su** <usuario>
- b **ovc -start**

Los procesos de agente y de servicio de control ahora se ejecutan como el usuario especificado.

Cambie el usuario predeterminado de comandos

De manera predeterminada, el agente inicia comandos automáticos o iniciados por el operador en la cuenta del usuario en la que se está ejecutando el mismo agente. Sin embargo, se puede configurar un HP Operations agent para que inicie los comandos en otra cuenta de usuario. Para ello, configure el parámetro `OVO_STD_USER` en el espacio de nombre `eaagt` de los nodos. Este parámetro se puede configurar de las siguientes formas:

- Configure los valores del ajuste predeterminado de la instalación de HP Operations agent. Esto se recomienda si es preciso configurar el usuario para un gran número de nodos. Hay que planificar y configurar los valores predeterminados de la instalación antes de crear o migrar los nodos.
- Use `ovconfchg` u `ovconfpar` en el símbolo del sistema.

Especifique el valor de `OVO_STD_USER` en el formato <usuario>/<palabra_clave_cifrada>

- Sustituya <usuario> por el nombre del usuario. En usuarios de dominio, especifique el dominio y nombre del usuario, por ejemplo: `EXAMPLE\AgentUser`. En usuarios locales, especifique sólo el nombre, por ejemplo `AgentUser`.
- Sustituya <palabra_clave_cifrada> por salida del comando `opcpwcrpt` <palabra_clave>. Puede iniciar este comando desde un símbolo del sistema de comando en el servidor de administración.

También puede usar `OVO_STD_USER` al configurar o iniciar una herramienta. Especifique el nombre del usuario `$OVO_STD_USER` y deje la palabra clave en blanco.

Deberá comprobar si la cuenta de usuario tiene los derechos necesarios para ejecutar comandos y herramientas correctamente.

- ▶ Si no le es posible al agente iniciar un comando o herramienta como `OVO_STD_USER`, podrá iniciar el comando o herramienta con la misma cuenta de usuario con la que el agente está siendo ejecutado. Ello puede ocurrir si, por ejemplo, especifica incorrectamente un usuario o palabra clave.

4 Uso del programa utility

El programa `utility` es una herramienta que permite administrar y notificar información sobre archivos de registro, el archivo de parámetros de recopilación (`parm`) y el archivo de definiciones de alarmas (`alarmdef`). El programa `utility` puede utilizarse de manera interactiva o en modo por lotes para realizar las tareas siguientes:

- Explorar archivos sin formato o archivos de registro extraídos y crear un informe que muestre:
 - cobertura de fechas y horas
 - horas en las que el recopilador `scope` no ha estado ejecutándose
 - cambios en la configuración de parámetros de `scope`
 - cambios en la configuración del sistema
 - espacio del disco de archivos de registro
 - efectos de la configuración del proceso y aplicación en el archivo de parámetros de recopilación (`parm`)
- Cambiar el tamaño de los archivos de registro sin formato
- Comprobar si el archivo `parm` contiene errores o advertencias de sintaxis
- Comprobar si el archivo `alarmdef` contiene errores o advertencias de sintaxis
- Procesar datos de archivo de registro con definiciones de alarmas para detectar condiciones de alarma en datos históricos

Este capítulo aborda los temas siguientes:

- [Ejecución del programa utility](#)
- [Uso del modo interactivo](#)
- [Uso de la Interfaz de línea de comandos del programa utility](#)
- [Detalles del informe de exploración de utility](#)

Pueden encontrarse descripciones detalladas de los comandos del programa `utility` en el [Capítulo 5, Comandos del programa utility](#).

Ejecución del programa `utility`

Existen tres formas de ejecutar el programa `utility`:

- Modo de línea de comandos - el usuario controla el programa `utility` por medio de opciones de comandos y argumentos en la línea de comandos.
- Modo interactivo - El usuario proporciona parámetros y comandos interactivos al ejecutar el programa con `stdin` conectado a una terminal o estación de trabajo interactiva. Los usuarios experimentados podrán especificar fácilmente sólo los comandos necesarios para realizar una tarea determinada. En el caso de usuarios nuevos se recomienda el uso del comando de la guía del programa `utility` para obtener asistencia acerca del uso de los comandos. En el modo guiado se solicita al usuario que seleccione una opción de una lista de opciones para realizar una tarea. En el caso del modo guiado, los comandos interactivos que permiten realizar cada una de las tareas se muestran en una lista conforme son ejecutados, de manera que el usuario pueda ver cómo se usan. Es posible salir y volver a entrar en el modo guiado en cualquier momento.
- Modo por lotes - El usuario puede ejecutar el programa y redirigir `stdin` a un archivo que contiene comandos y parámetros interactivos.

La sintaxis correspondiente a la interfaz de línea de comandos es similar a las interfaces de línea de comandos UNIX convencionales y otros programas, y se describe en detalle en este capítulo.

En el caso del modo interactivo y por lotes, la sintaxis de comandos es la misma. Los comandos pueden introducirse en cualquier orden. Si el comando ha sido asociado a un parámetro en particular, dicho parámetro deberá ser introducido inmediatamente después del comando correspondiente.

Existen dos tipos de parámetros: los requeridos (para los cuales no existen predeterminados), y los opcionales (para los cuales se ofrecen predeterminados). El modo en que `utility` gestione estos parámetros dependerá del modo en el que se ejecute.

- En el modo interactivo, si falta un parámetro opcional, el programa mostrará el argumento predeterminado y dejará que el usuario lo confirme o invalide. Si falta un parámetro requerido, el programa solicitará al usuario que introduzca el argumento.
- En el modo por lotes, si falta un parámetro opcional, el programa usará los valores predeterminados. Si falta un parámetro requerido, el programa finalizará.

En el modo interactivo los errores y la falta de datos son tratados de manera distinta que en el modo por lotes y en el modo de línea de comandos. El usuario podrá proporcionar datos adicionales o corregir errores en el modo interactivo, pero no en el modo por lotes o en el modo de línea de comandos.

Uso del modo interactivo

Para utilizar el modo interactivo del programa `utility` es necesario emitir una serie de comandos para ejecutar una tarea en particular.

Por ejemplo, si se desea comprobar si en un archivo de registro existen condiciones de alarma en datos registrados durante el día en curso, el usuario deberá emitir los siguientes comandos para invocar el programa `utility`:

```
checkdef /var/opt/perf/alarmdef  
detail off  
start today-1  
analyze
```

El comando `checkdef` comprueba la sintaxis de las definiciones de alarma en el archivo `alarmdef` y, a continuación, configura y guarda el nombre del archivo para su uso con el comando `analyze`. El comando `detail off` se ocupa de que el comando `analyze` muestre sólo un resumen de las alarmas. El comando `start today-1` especifica que sólo serán analizados los datos registrados el día de ayer. El comando `analyze` analiza los archivos de registro sin formato en el origen de datos SCOPE predeterminado contrarrestándolos con el archivo `alarmdef`.

Ejemplo de uso del modo interactivo y el modo por lotes

El siguiente ejemplo muestra las diferencias entre el modo de funcionamiento del comando `resize` del programa `utility` en el modo por lotes y en el modo interactivo.

El comando `resize` permite establecer los parámetros de las funciones siguientes:

- Tipo de archivo de registro a ser cambiado de tamaño.
- Tamaño del nuevo archivo.
- Cantidad de espacio libre a dejar en el archivo.
- Acción que especifica si se realizará o no el cambio de tamaño.

Este ejemplo de comando `resize` modifica el tamaño del archivo de registro global de manera que contenga un máximo de 120 días de datos con espacio libre equivalente a 45 días. El comando y sus parámetros correspondientes son:

```
resize global days=120 empty=45 yes
```

Los resultados son los mismos, independientemente de si se introduce este comando de manera interactiva o por medio de un trabajo por lotes.

El primer parámetro `global` indica el archivo de registro a ser cambiado de tamaño. Si no se proporciona este parámetro, la acción consiguiente para tanto usuarios interactivos como para usuarios por lotes será la siguiente:

- Usuarios por lotes — el trabajo por lotes finalizará porque el parámetro `logfile` no tiene valor predeterminado.
- Usuarios interactivos — aparecerá un mensaje para solicitar al usuario el tipo de archivo de registro que desea cambiar de tamaño para finalizar el comando.

El último parámetro `yes` indica que el cambio de tamaño se realizará incondicionalmente.

Si no se proporciona el parámetro `yes`, la acción consiguiente para tanto usuarios interactivos como para usuarios por lotes será la siguiente:

- Usuarios por lotes — el cambio de tamaño continuará ya que `yes` es la acción predeterminada.
- Usuarios interactivos — se solicitará al usuario que proporcione la acción antes de que tenga lugar el cambio de tamaño.



Antes de usar el comando `cambiar tamaño` bien en el modo por lotes o en el modo interactivo, deberá detenerse la recopilación de datos. Para obtener mayor información, consulte [Detención y reinicio de la recopilación de datos](#) en la página 40 en el capítulo 2.

Interfaz de línea de comandos del programa `utility`

Además de la sintaxis de comandos para el modo interactivo y el modo por lotes, las opciones de comando y sus argumentos asociados pueden transferirse al programa `utility` por medio de la interfaz de línea de comandos. La interfaz de línea de comandos funciona bien en entornos UNIX convencionales ya que permite que el programa `utility` pueda ser invocado fácilmente por secuencias de comandos shell posibilitando que tanto las entradas como las salidas sean redireccionadas a canalizaciones UNIX.

Por ejemplo, para usar el equivalente de la línea de comandos del ejemplo mostrado en la sección anterior "Uso del modo interactivo", introduzca lo siguiente:

```
utility -xr global days=120 empty=45 yes
```

Los argumentos y opciones de línea de comandos se enuncian en la tabla siguiente. Las descripciones de comandos referenciados se encuentran en el [Capítulo 5, Comandos del programa `utility`](#).

Tabla 3 Argumentos de línea de comandos

Opción de comando	Argumento		Descripción
-b	date	time	Especifica la fecha de inicio y la hora de una función de exploración o de análisis. (Consulte el comando start en el capítulo 4).
-e	date	time	Especifica la fecha y la hora término de una función de exploración o de análisis. (Consulte el comando stop en el capítulo 4).
-l	logfile		Especifica el archivo de registro a abrir. (Consulte el comando logfile en el capítulo 4).
-f	listfile		Especifica un archivo de listado de salidas. (Consulte el comando list en el capítulo 4).
-D			Habilita los detalles para la comprobación de los archivos <code>analyze</code> , <code>scan</code> y <code>parm</code> . (Consulte el comando detail en el capítulo 4).
-d			Deshabilita los detalles para la comprobación de los archivos <code>analyze</code> y <code>parm</code> . (Consulte el comando detail en el capítulo 4).

Tabla 3 Argumentos de línea de comandos

Opción de comando	Argumento	Descripción	
-v		Reproduce los comandos de la línea de comandos conforme son ejecutados.	
-xp	parmfile	La sintaxis comprueba un archivo parm. (Consulte el comando parmfile en el capítulo 4).	
-xc	alarmdef	La sintaxis comprueba y establece el nombre del archivo alarmdef para su uso con -xa (o el comando analyze). (Consulte el comando checkdef en el capítulo 4).	
-xa		Analiza archivos de registro contrastándolos con el archivo alarmdef. (Consulte el comando Comando analyze en el capítulo 4).	
-xs	logfile	Explora un archivo de registro y crea un informe. (Consulte el comando scan en el capítulo 4).	
-xr	global application process device transaction ls EMPTY=nnn SPACE=nnn	SIZE=nn n DAYS=nn n YES NO MAYBE	Cambia el tamaño de un archivo de registro. (Consulte el comando resize en el capítulo 4).
-? or ?		Muestra la sintaxis de la línea de comando.	

Ejemplo de uso de la interfaz de línea de comandos

La siguiente situación se aplica cuando se introducen argumentos y opciones de comandos en la línea de comandos:

Los errores y la falta de datos son tratados exactamente igual que en el comando de modo por lotes correspondiente. Es decir, faltan datos se convierte en la acción predeterminada si es posible y todos los errores hacen que el programa finalice de inmediato.

La opción reproducción de comandos y los resultados de comandos es deshabilitada. `Utility` no puede leer de su archivo `stdin`. Finaliza después de las acciones de la línea de comandos.

```
utility -xp -d -xs
```

Que se traduce en:

- xp La sintaxis comprueba el archivo `parm` predeterminado.
- d Deshabilita los detalles en el informe de exploración.
- xs Realiza la operación de `scan`. No se ha especificado ningún archivo de registro y por lo tanto el archivo de registro predeterminado es explorado.

Detalles del informe de exploración de utility

El comando `scan` del programa `utility` lee un archivo de registro y escribe un informe sobre su contenido. El contenido de los informes dependerá de los comandos emitidos con anterioridad a la emisión del comando `scan`. (Para más información, consulte la descripción del comando `scan` en el [Capítulo 5, Comandos del programa utility](#)).

La tabla siguiente resume la información contenida en los informes de exploración y en los informes que sólo se crean cuando se usa el comando `detail on` (el predeterminado) con el comando `scan`.

Tabla 4 Información contenida en el informe de exploración

Valores iniciales	
Información global inicial del archivo <code>parm</code> e información de configuración del sistema	Impreso sólo si se especifica <code>detail on</code> .
Definiciones de aplicación iniciales del archivo <code>parm</code>	Impreso sólo si se especifica <code>detail on</code> .
Detalles cronológicos	
Cambios globales del archivo <code>parm</code>	Impreso sólo si se especifica <code>detail on</code> .
Cambios de aplicación del archivo <code>parm</code>	Impreso sólo si se especifica <code>detail on</code> .

Tabla 4 Información contenida en el informe de exploración

Notificaciones de tiempo de inactividad del recopilador	Impreso sólo si se especifica <code>detail on</code> .
Informes de resúmenes específicos de aplicación	Impreso sólo si se especifica <code>detail on</code> .
Resúmenes	
Informe de resumen de proceso	Impreso siempre que los datos de proceso sean explorados.
Resumen de cobertura del recopilador	Impreso siempre.
Resumen de contenidos del archivo de registro	Impreso siempre. Incluye espacio y fechas de cobertura.
Resumen de espacio libre del archivo de registro	Impreso siempre.

Información del informe de exploración

La información contenida en el informe de exploración de `utility` se divide en tres tipos:

- Valores iniciales
- Detalles cronológicos
- Resúmenes

Valores iniciales

En esta sección se describen los valores iniciales siguientes:

- Información global inicial del archivo `parm`
- Definiciones de aplicación iniciales del archivo `parm`

Información global inicial del archivo `Parm`

Para obtener este informe, use el comando `scan` con su `detail on` predeterminado.

Este informe indica los parámetros de configuración del archivo `parm` existentes en el momento en que se registra el primer registro global en el archivo de registro. Las notificaciones de cambio en información global posteriores se basarán en los valores de este informe. Si no se produce ninguna notificación de cambio respecto a un parámetro en particular, ello significará que el parámetro mantiene su configuración original durante el proceso de exploración.

El ejemplo siguiente muestra un fragmento de un informe en el que se indica el contenido del archivo parm.

```
06/03/99 15:28 System ID="Homer"
scopeux/UX A.10.00 SAMPLE INTERVAL = 300,300,60 Seconds, Log version=D
Configuration: 9000/855, O/S A.10.00 CPUs=1
Logging Global Process records
      Device= Disk FileSys records
Thresholds: CPU= 10.00%, Disk=10.0/sec, First=5.0 sec, Resp=30.0 sec,

      Trans=100 Nonew=FALSE, Nokilled=FALSE, Shortlived=FALSE
      (<1 sec)
HP-UX Parms: Buffer Cache Size = 16384KB, NPROC = 532
Wait Thresholds: CPU=100.00%, Memory=100.00%
Impede=100.00%
Memory: Physical = 84.0 MB, Swap = 124304.0 MB, Available to users = 66.5
MB. There are 2 LAN interfaces: 0, 1.
06/03/99 15:28 There are 2 disk devices:
      Disk #1976          = "/dev/hdisk0"
      Disk #1987          = "/dev/hdisk1"
```

La fecha y hora indicadas en la primera línea corresponden a la *primera fecha y hora* del archivo de registro global e indican cuándo se inició scope. Los registros de datos pueden haber sido descartados del archivo de registro global, de manera que la fecha y hora de este informe no necesariamente indican el *primer registro global* del archivo de registro.

Definiciones iniciales de aplicación del archivo parm

Para obtener este informe, use el comando scan con su detail on predeterminado y asegúrese de tener datos de aplicación en el archivo de registro.

Este informe indica el nombre y la definición de cada aplicación en el momento en que el primer registro de aplicación es incluido en el archivo de registro. Toda notificación de adición o eliminación de aplicación que se reciba estará basada en esta lista inicial de aplicaciones. Por ejemplo:

```
06/01/99 08:39 Application(1) = "other"
Comment=all processes not in user-defined applications

06/01/99 08:39 Application(2) = "Real_TimeSystem"
Priority range = 0-127

06/01/99 08:39 Application(3) = "Prog_Development"
File=vi,ed,ed,xd,ld,lint,cc,ccom,pc,pascomp
```



Durante la exploración, el usuario será notificado de las aplicaciones añadidas o eliminadas. Las adiciones o eliminaciones se determinan comparando la ortografía y las mayúsculas/minúsculas de los nombres de aplicación con el nuevo conjunto de nombres de aplicación registrados. No se ha contemplado en ningún momento detectar cambios en la definición de una aplicación. Si se detecta una aplicación con un nombre nuevo, éste será incluido en la lista junto con su nueva definición.

La fecha y hora del registro se corresponde con la última vez que scope fue iniciado antes de registrar el primer registro de aplicación existente en el archivo de registro.

Detalles cronológicos

En esta sección se describen los detalles cronológicos siguientes:

- notificaciones de cambios globales del archivo `parm`
- Notificaciones de adiciones y eliminaciones de aplicaciones en el archivo `parm`.
- Notificaciones de tiempo de inactividad de `scope`.
- Informe de resumen específico de la aplicación.

Notificaciones de cambios globales del archivo `parm`

Para obtener este informe, use el comando `scan` con su `detail` on predeterminado.

Este informe es generado en cualquier momento en que un registro detecte que `scope` se ha iniciado.

El siguiente ejemplo muestra las notificaciones de cambio que tienen lugar cuando dos unidades de disco nuevas se añaden al sistema.

```
03/13/99 17:30 The number of disk drives changed from 9 to 11
03/13/99 17:30 New disk device scsi-4 = "c4d0s*"
03/13/99 17:30 New disk device scsi-3 = "c3d0s*"
```

Notificaciones de adiciones/eliminaciones de aplicaciones en el archivo `parm`

Para obtener este informe, use el comando `scan` con su `detail` on predeterminado y asegúrese de tener datos de aplicación en el archivo de registro.

Se pueden añadir o eliminar aplicaciones definidas por el usuario cada vez que se inicie `scope`. Si el nombre de una aplicación no se corresponde con el último conjunto de aplicaciones, se imprimirá una notificación de adición, eliminación o cambio. Si el nombre de la aplicación no ha cambiado, no se imprimirá nada.

El siguiente ejemplo muestra que se ha iniciado una nueva aplicación.

```
03/13/99 17:30 Application 4 "Accounting_Users_1" was added
User=ted,rebecca,test*,mark,gene
```



No se comprobará si han habido cambios en las definiciones de aplicación. Se indicarán en una lista si cambia el nombre de una aplicación, pero no se detectará ningún cambio realizado a la definición de una aplicación existente si no va acompañada de un cambio de nombre.

Notificaciones de tiempo de inactividad de `scope`

Para obtener este informe, use el comando `scan` con su `detail` on predeterminado.

Si un archivo extraído sólo contiene información de resumen, la hora será redondeada a la hora más próxima. Por ejemplo:

```
06/03/99 11:00 - 06/03/99 12:34 collector off (01:34:04)
```

La primera fecha y hora (06/03/99 11:00) indica el último registro de datos válido contenido en el archivo de registro antes de que se reiniciara `scope`. La segunda fecha y hora (06/03/99 12:34) indica cuándo ha sido reiniciado `scope`.

El último campo (en paréntesis) muestra el tiempo durante el cual *scope no ha sido* ejecutado. El formato es *ddd/hh:mm:ss*, en donde *ddd* son los días y *hh:mm:ss* son las horas, los minutos y los segundos. Los ceros a la izquierda son eliminados.

En este ejemplo, *scope* no fue ejecutado el 3 de junio de 1999 entre las 11:00 am y las 12:34 pm. El informe de resumen muestra que no se recopiló datos durante 1 hora, 34 minutos y 4 segundos.

Informe de resumen específico de la aplicación

Para obtener este informe, use el comando `scan` con su `detail` on predeterminado y asegúrese de tener datos de aplicación en el archivo de registro.

Este informe es útil para definir aplicaciones. Use el informe para identificar aplicaciones que estén acumulando o demasiados o escasos recursos del sistema y aplicaciones que puedan ser consolidadas con otras aplicaciones. Puede resultar útil dividir en múltiples aplicaciones aquellas aplicaciones que estén acumulando demasiados recursos del sistema.

Se aconseja que se definan las aplicaciones de manera que ayuden a tomar decisiones en materia de ajustes de rendimiento. Es improbable que los recursos del sistema se acumulen de manera homogénea entre aplicaciones.

El informe de resumen específico de la aplicación será generado cuando cambien las definiciones de la aplicación con objeto de permitir al usuario el acceso a los datos de las definiciones de la aplicación antes y después de que se produzca el cambio.

Se generará un informe final por aplicación. Este informe sólo comprende el tiempo transcurrido desde el último informe y no el tiempo total que cubre el archivo de registro. Por ejemplo:

Application	PERCENT OF TOTAL			
	Records	CPU	DISK	TRANS
-----	-----	-----	-----	-----
OTHER	22385	45.7%	20.9%	63.0%
Resource_Sharing	7531	6.0%	2.2%	17.1%
SPOOLING	13813	2.4%	0.3%	0.0%
ON_LINE_COMPILE	13119	2.9%	1.7%	0.1%
BATCH_COMPILE	8429	2.9%	0.1%	2.2%
ORDER_ENTRY	387	0.1%	0.0%	0.0%
ELECTRONIC_MAIL	6251	3.8%	1.3%	9.6%
PROGRAM_DEVELOPMENT	3141	9.1%	2.4%	0.6%
RESEARCH_DEPARTMENT	3968	8.7%	2.0%	6.0%
BILL_OF_MATERIALS	336	0.6%	1.5%	0.1%
FINANCIALS	1080	5.0%	1.5%	0.5%
MARKETING_DEPT	2712	12.9%	67.3%	0.0%
GAMES	103	0.1%	0.0%	0.0%
-----	-----	-----	-----	-----
All user applications	73.1%	54.3%	79.1%	37.0%

Resúmenes

En esta sección se describen los resúmenes siguientes:

- Resumen de motivos de registro de proceso
- Hora y fecha reales de inicio y parada de exploración
- Resumen general de aplicación
- Resumen de cobertura de `scope`
- Resumen de contenidos del archivo de registro
- Resumen de espacio libre del archivo de registro

Resumen de motivos de registro de proceso

Para obtener este informe se deberán tener datos de proceso en el archivo de datos.

Este informe es útil para establecer los umbrales de proceso interesantes de `scope`. Este informe indica cada uno de los motivos por los que un proceso puede considerarse interesante y, por lo tanto, es registrado, junto con el número total de procesos registrados que cumplen cada condición.

El ejemplo siguiente muestra un informe de resumen de motivos de registro de proceso:

```
Process Summary Report: 04/13/99 3:32 PM to 05/04/99 6:36 PM
There were 93.8 hours of process data
Process records were logged for the following reasons:
```

Log Reason	Records	Percent	Recs/hr
-----	-----	-----	-----
New Processes	17619	53.9%	44.7
Killed Processes	16047	49.1%	40.7
CPU Threshold	3169	9.7%	8.0
Disk Threshold	1093	3.3%	2.8

NOTE: A process can be logged for more than one reason at a time. Record counts and percentages will not add up to 100% of the process records.

Si el comando `detail on` es emitido, este informe se generará cada vez que se cambie un valor de umbral de forma que se puedan evaluar los efectos de dicho cambio. Cada informe cubre el periodo transcurrido desde el último informe. Un último informe que es generado cuando finaliza la exploración cubre el tiempo transcurrido desde el último informe.

Si el comando `detail off` es emitido, sólo se generará un informe que cubre el periodo explorado en su totalidad.

Se puede reducir la cantidad de datos de proceso registrados por `scope` modificando el parámetro `threshold` del archivo `parm` y aumentando los umbrales de los motivos de interés que generan el máximo de registros de registro de procesos. Para incrementar la cantidad de datos registrados, bájese el umbral del área de interés.

En el ejemplo anterior se podía disminuir la cantidad de espacio de disco usado por los datos de proceso (a costa de tener menos información registrada) aumentando el umbral de CPU o estableciendo el umbral `nonew`.

Parada e inicio de exploración

Este informe de resumen se imprimirá si se han explorado datos válidos. Ofrece las fechas y horas reales en las que se se inició y detuvo la exploración. Por ejemplo:

```
Scan started on      03/03/99 12:40 PM
Scan stopped on     03/11/99  1:25 PM
```

Resumen general de aplicación

Para obtener este informe se deberán tener datos de aplicación en el archivo de datos.

Este informe es un indicador general de la cantidad de actividad del sistema que está siendo acumulada en aplicaciones definidas por el usuario, en lugar de en la otra aplicación. Si no está siendo capturada una cantidad significativa de un recurso crítico por aplicaciones del usuario, se deberá considerar la posibilidad de explorar los datos de proceso para detectar procesos que puedan incluirse en aplicaciones de usuario.

Por ejemplo:

```
OVERALL, USER DEFINED APPLICATIONS ACCOUNT FOR
82534 OUT OF      112355 RECORDS      ( 73.5%)
218.2 OUT OF      619.4 CPU HOURS      ( 35.2%)
24.4 OUT OF       31.8 M DISC IOS      ( 76.8%)
0.2 OUT OF        0.6 M TRANS        ( 27.3%)
```

Resumen de cobertura del recopilador

Este informe se imprimirá si se han explorado datos de aplicación o datos globales válidos. Indica el grado de eficiencia con el que se está usando `scope` para recopilar la actividad del sistema. Si el porcentaje de tiempo que `scope` ha estado inactivo es alto, como en el ejemplo que se muestra a continuación, deberán revisarse los procesos operativos relativos a la parada e inicio de `scope`.

```
The total time covered was      108/16:14:51 out of 128/00:45:02
Time lost when collector was off 19/08:30:11 15.12%
The scopeux collector was started 45 times
```

Este informe será más completo si los datos de detalles globales se incluyen en la exploración. Si sólo se disponen de los datos de resumen, el usuario sólo podrá determinar el tiempo en que `scope` ha estado parado y se ha iniciado redondeado a la hora más próxima. (Se imprimirá un mensaje de advertencia con el informe si esto ocurre).

El tiempo total cubierto se determina acumulando todos los intervalos horarios de los datos registrados. El valor de tiempo "out of" se calcula restando la hora y fecha finales por la hora y fecha de inicio. Ello deberá representar el tiempo total que podría haber sido registrado. El valor "Time lost when collector was off" es igual al tiempo total menos el tiempo cubierto.

Los formatos correspondientes a los tres tiempos aquí mencionados son:

ddd/hh:mm:ss

en donde *ddd* son los días y *hh:mm:ss* son las horas, minutos y segundos.

En el ejemplo anterior el tiempo total recopilado fue de 108 días, 16 horas, 14 minutos y 51 segundos.

Resumen de contenidos del archivo de registro

El resumen de contenidos del archivo de registro se imprimirá *si* se exploran datos válidos. Ello incluye el espacio de archivo registrado y las fechas cubiertas. Este resumen es útil al cambiar el tamaño de los archivos de registro mediante el comando `resize`.

-----Total-----	-----Each Full Day-----		-----Dates-----		Full		
Type	Records	MBytes	Records	MBytes	Start	Finish	Days
Global	1376	0.27	288.9	0.057	05/23/99	to 05/28/99	4.8
Application	6931	0.72	1455.0	0.152	05/23/99	to 05/28/99	4.8
Process	7318	1.14	1533.6	0.239	05/23/99	to 05/28/99	4.8
Disk	2748	0.07	567.6	0.014	05/23/99	to 05/28/99	4.8
Transaction	no data found						
Overhead		0.29					
TOTAL	18373	2.49	3845.0	0.461			

Las columnas se describen del siguiente modo:

Columna Explicación

Type	Tipo general de datos que están siendo registrados. Existe un tipo especial, Overhead: Overhead (sobrecarga) es la cantidad de espacio de disco ocupado (o reservado) por el archivo de registro <i>frente a</i> la cantidad realmente usada por los registros de datos explorados. Si se explora menos del archivo de registro completo, Overhead incluirá los registros de datos que no han sido explorados. Si el archivo completo es explorado, Overhead considerará toda ineficiencia ocurrida al bloquear los datos en el archivo y <i>también</i> todas las estructuras de soporte de acceso a archivos. En el caso de los archivos de registro extraídos es normal que tengan una sobrecarga mayor que los archivos de registro sin formato ya que presentan unas estructuras de soporte adicionales para un posicionamiento más rápido.
Total	Recuento total de registros y espacio de disco explorado por cada tipo de datos.
Each Full Day	Número de registros y cantidad de espacio de disco usada por cada periodo de 24 horas que se ejecuta <code>scope</code> .
Dates	Primeras y últimas fechas válidas correspondientes a los registros de cada tipo de datos explorado.
Full Day	Número de días (24 horas) completos de datos explorados correspondiente a este tipo de datos. Full Days puede no equivaler a la diferencia entre la fecha de inicio y de parada si la cobertura de <code>scope</code> no ha sido igual al 100 por ciento del tiempo explorado.

La línea TOTAL (debajo de los datos listados) ofrece una idea de cuánto espacio de disco se está utilizando y cuántos datos se esperan que se acumulen diariamente.

Resumen de espacio libre del archivo

Este informe se imprimirá con cada archivo de registro explorado. Por ejemplo:

```
The Global      file is now 13.9% full with room for 61 more full days
The Application file is now 15.1% full with room for 56 more full days
The Process     file is now 23.5% full with room for 32 more full days
The Device      file is now  1.4% full with room for 2896 more full days
```

La cantidad de espacio disponible para datos adicionales será calculada en base a los factores siguientes:

- La cantidad de espacio no utilizado en el archivo.
- El valor explorado correspondiente al número de megabytes de datos que está siendo registrado por día de 24 horas.

Si los valores correspondientes a los megabytes explorados por día resultan ser inexplicablemente bajos, serán reemplazados por los valores predeterminados de este cálculo.

Si se explora un archivo extraído se obtendrá una línea de informe única porque todos los tipos de datos comparten un mismo archivo extraído.

5 Comandos del programa utility

El presente capítulo describe los comandos del programa `utility`. Incluye un resumen de sintaxis y una sección de referencia de comandos que enumera los comandos en orden alfabético.

Los comandos y parámetros `Utility` pueden ser introducidos en cualquier combinación de minúsculas y mayúsculas. Sólo se requieren las tres primeras letras del nombre del comando. Por ejemplo, el comando `logfile` puede introducirse como **logfile** o puede abreviarse como **log** o **LOG**.

Para obtener ejemplos de cómo usar estos comandos consulte la ayuda en línea referente al programa `utility`.

La tabla que se muestra en las páginas siguientes contiene un resumen de los parámetros y sintaxis de comandos `utility`.

Tabla 5 Comandos utility: Sintaxis y parámetros

Comando	Parámetro
<code>analyze</code>	
<code>checkdef</code>	<code>archivo alarmdef</code>
<code>detail</code>	<code>on</code> <code>off</code>
<code>exit</code> <code>e</code>	
<code>guide</code>	
<code>list</code>	<i>nombre de archivo o *</i>
<code>logfile</code>	<code>logfile</code>
<code>menu</code> <code>?</code>	
<code>parmfile</code>	<code>parmfile</code>
<code>quit</code> <code>q</code>	

Tabla 5 Comandos utility: Sintaxis y parámetros

Comando	Parámetro
resize	global application process device transaction days=maxdays size=max MB empty=days space=MB yes no maybe
scan	<i>archivo de registro</i> (la operación también puede verse afectada por los comandos list, start, stop y detail.
show	all
sh !	comando de sistema
start	<i>fecha [hora]</i> today [-días] [hora] last [-días] [hora] first [+días] [hora]
stop	<i>fecha [hora]</i> today [-días] [hora] last [-días] [hora] first [+días] [hora]

Comando analyze

El comando `analyze` permite analizar los datos contenidos en un archivo de registro contrarrestándolos con las definiciones de alarma de un archivo de definiciones de alarma (`alarmdef`) y notificar la actividad y el estado de la alarma resultantes. Se recomienda que se ejecute el comando `checkdef` antes de emitir el comando `analyze` para comprobar la sintaxis de definiciones de alarma. `checkdef` también establece y guarda el nombre de archivo de definiciones de alarma a ser usado con `analyze`. Si no se ejecuta `checkdef` antes de `analyze`, se solicitará al usuario un nombre de archivo de definiciones de alarma.

Si se usa el modo de línea de comandos, se usará el archivo de definiciones de alarma predeterminado `/var/opt/perf/alarmdef`.

Para obtener información detallada sobre definiciones de alarmas, consulte [Alarmas de rendimiento](#) en la página 161.

Sintaxis

`analyze`

Instrucciones sobre su uso

Al usar el comando `analyze`, éste analiza los archivos de registro especificados en el archivo de configuración de orígenes de datos, `datasources`, contrastándolos con las definiciones de alarma del archivo `alarmdef`.

El comando `analyze` permite evaluar si una definición de alarma se corresponde o no con los datos históricos recopilados en un sistema. Permite además al usuario decidir si sus definiciones de alarma generarán muchas o pocas alarmas en su estación de trabajo de análisis.

Asimismo, el usuario podrá realizar análisis de datos con definiciones (instrucciones IF) establecidas en el archivo de definiciones de alamar porque le será posible obtener salida de información con las instrucciones PRINT cuando se cumplan las condiciones. Para una explicación de cómo usar las instrucciones IF y PRINT en una definición de alarma, consulte el [Capítulo 9, Alarmas de rendimiento](#).

El usuario podrá ejecutar de manera opcional los comandos `start`, `stop` y `detail` con `analyze` para personalizar el proceso de análisis. Los comandos se especificarán en el orden siguiente:

```
checkdef
start
stop
detail
analyze
```

Los comandos `start` y `stop` permiten analizar datos de archivo de registro que han sido recopilados durante un periodo de tiempo determinado. (Más adelante en este capítulo se describen los comandos `start` y `stop`).

Mientras se ejecuta el comando `analyze`, éste indicará eventos de alarma tales como `alarm start`, `end` y `repeat status` junto con el texto contenido en instrucciones de impresión asociadas. Además, todo texto contenido en las instrucciones PRINT será mostrado conforme se cumplen las condiciones (en las instrucciones IF). Las instrucciones EXEC no serán ejecutadas pero sí mostradas de manera que el usuario pueda ver lo que habría sido ejecutado. Un informe de resumen de alarmas mostrará un recuento del número de alarmas y del tiempo total en que cada alarma ha estado activa (`on`). El recuento incluye `alarm starts` y `repeats`, pero no `alarm ends`.

Para sólo ver el informe de resumen de alarmas, debe emitirse el comando `detail off`. Sin embargo, si se está usando el modo de línea de comandos, `detail off` será el comando predeterminado, por lo que se necesitará especificar `-D` para poder ver tanto los eventos de alarma como el resumen de alarmas.

Ejemplo

El comando `checkdef` comprueba la sintaxis de las definiciones de alarma en el archivo `alarmdef` y, a continuación, guarda el nombre del archivo `alarmdef` para su uso posterior con el comando `analyze`. El comando `start today` especifica que sólo serán analizados los datos registrados el día de hoy. Por último, el comando `analyze` analiza el archivo de registro en el origen de datos predeterminado `SCOPE` especificado en el archivo `datasources` contrastándolo con las definiciones de alarma del archivo `alarmdef`.

```
utility>
checkdef /var/opt/perf/alarmdef
start today
analyze
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
utility -xc -D -b today -xa
```

checkdef

El comando `checkdef` permite comprobar la sintaxis de las definiciones de alarma en el archivo de definiciones de alarma y notificar toda advertencia o error que sea detectado. Este comando también establece y guarda el nombre del archivo de definiciones de alarma para su uso con el comando `analyze`.

Para obtener descripciones de la sintaxis de definiciones de alarma e información sobre cómo especificar las definiciones de alarma, consulte el [Capítulo 9, Alarmas de rendimiento](#).

Sintaxis

```
checkdef [/directorypath/alarmdef]
```

Parámetros

`alarmdef` El nombre de cualquier archivo de definiciones de alarma. Éste puede ser el archivo especificado por el usuario o el archivo `alarmdef` predeterminado. Si no se especifica la ruta de acceso de directorio, se buscará el directorio actual.

Instrucciones sobre su uso

Una vez que se haya determinado que las definiciones de alarma son correctas, se podrán procesar contrastándolas con los datos contenidos en un archivo de registro usando el comando `analyze`.

En el modo por lotes, si no se ha especificado ningún archivo de definiciones de alarma, se usará el archivo predeterminado `alarmdef`.

En el modo interactivo, si no se ha especificado ningún archivo de definiciones de alarma, se solicitará que se especifique uno.

Ejemplo

El comando `checkdef` comprueba la sintaxis de las definiciones de alarma en el archivo `alarmdef` y, a continuación, guarda el nombre del archivo `alarmdef` para su uso posterior con el comando `analyze`.

```
utility>
checkdef /var/opt/perf/alarmdef
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
utility -xc
```

detail

El comando `detail` permite controlar el nivel de detalles imprimidos en los informes `analyze`, `parmfile` y `scan`.

El valor predeterminado es `detail on` en los modos interactivo y por lotes y `detail off` en el modo de línea de comandos.

Sintaxis

```
detail      [on]
              [off]
```

Parámetros

- `on` Imprime el contenido del archivo `parm` así como los errores del archivo `parm`. Imprime informes completos de `analyze` y `scan`.
- `off` En el informe del archivo `parm`, las definiciones de aplicación *no* son imprimidas. En el informe `scan`, no son imprimidos ni la hora de recopilación de `scope`, ni la información global inicial del archivo `parm`, ni las definiciones de aplicación. En el informe `analyze`, los eventos de alarma y las acciones de alarma *no* son imprimidos.

Instrucciones sobre su uso

Para obtener instrucciones sobre el uso del comando `detail` en combinación con los comandos `analyze`, `scan` y `parmfile`, consulte las descripciones de [Comando `analyze`](#), [parmfile](#) y [scan](#) contenidas en este capítulo.

Ejemplos

Para obtener ejemplos del uso del comando `detail`, consulte las descripciones de [Comando `analyze`](#), [parmfile](#) y [scan](#) contenidas en este capítulo.

exit

El comando `exit` permite finalizar el programa `utility`. El comando `exit` equivale al comando `quit` del programa `utility`.

Sintaxis

```
exit  
e
```

guide

El comando `guide` permite introducir el modo de comandos guiados. La interfaz de comandos guiados conduce al usuario por los diversos comandos `utility` y solicita al usuario que realice las tareas más comunes que estén disponibles.

Sintaxis

```
guide
```

Instrucciones sobre su uso

- Para introducir el modo de comandos guiado en el modo interactivo de `utility`, el usuario deberá escribir **guide** y pulsar **Entrar**.
- Para aceptar el valor predeterminado correspondiente a un parámetro, el usuario deberá pulsar **Entrar**.
- Para finalizar el modo de comandos guiados y regresar al modo interactivo, el usuario deberá escribir **q** tras el aviso `guide>`.

Este comando no ofrece todas las combinaciones posibles de configuración de parámetros. Selecciona la configuración que debería producir resultados útiles para la mayoría de los usuarios.

help

El comando `help` permite acceder a la utilidad de ayuda en línea del programa `utility`.

Sintaxis

```
help [palabra clave]
```

Instrucciones sobre su uso

Pueden introducirse parámetros para obtener información sobre las tareas y los comandos `utility`, o en `help` mismo. Se puede navegar por los diferentes temas introduciendo una palabra clave. Si se encuentra disponible más de una página de información, la pantalla se detiene y se espera a que el usuario pulse **Entrar** para continuar. Escriba **q** o **quit** para salir del sistema de ayuda y regresar al programa `utility`.

También se puede pedir ayuda sobre un tema determinado. Por ejemplo,

```
help tasks
```

o bien

```
help resize parmfile
```

Cuando se use este modo del comando `help`, el usuario recibirá el texto de ayuda para el tema especificado y permanecerá en el contexto de entrada de comando `utility`. Como no se introduce el subsistema de ayuda de forma interactiva, no se tiene que escribir **quit** antes de introducir el siguiente comando `utility`.

list

El comando `list` permite especificar el archivo de salida de todos los informes `utility`. El contenido del informe dependerá de si se emiten otros comandos después del comando `list`. Por ejemplo, el uso del comando `list` antes de los comandos `logfile`, `detail on` y `scan` produce el archivo de lista correspondiente a un informe de resumen detallado de un archivo de registro.

Sintaxis

```
list [nombre de archivo] | *
```

en donde `*` vuelve a establecer la salida en `stdout`.

Instrucciones sobre su uso

Existen dos modos de especificar el archivo `list` para informes:

- Redireccionar `stdout` cuando se invoca el programa `utility` escribiendo:

```
utility > utilrept
```

- O, usar el comando `list` cuando está ejecutándose `utility` escribiendo:

```
list utilrept
```

En ambos casos, los errores e interacciones del usuario se imprimen en `stderr` y los informes van al archivo especificado.

El parámetro *nombre de archivo* del comando `list` debe representar un nombre de archivo válido al cual el usuario tiene acceso de escritura. Los archivos existentes tienen la nueva salida anexa al final de los contenidos existentes. Si el archivo no existe, será creado.

Para determinar el archivo de salida actual, emita el comando `list` sin parámetros:

Si el archivo de salida no es `stdout`, la mayoría de los comandos serán reproducidos en el archivo de salida conforme sean introducidos.

Ejemplo

El comando `list` produce un informe de resumen en el archivo de registro extraído `rxlog`. El comando `list utilrept` direcciona el listado de informes de exploración a un archivo de disco. `Detail off` especifica menos de totalidad de detalles en el informe. El comando `scan` lee `rxlog` y produce el informe.

El comando `list *` vuelve a establecer el dispositivo de listas al `stdout` predeterminado. `!lp utilrept` envía el archivo de disco a la impresora del sistema.

```
utility>
logfile rxlog
list utilrept
detail off
scan
list *
!lp utilrept
```

Para realizar la tarea anterior usando los argumentos de la línea de comando, introduzca:

```
utility -l rxlog -f utilrept -d -xs print utilrept
```

logfile

El comando `logfile` permite abrir un archivo de registro. Para muchas funciones del programa `utility` debe encontrarse abierto un archivo de registro. Ello puede hacerse explícitamente emitiendo el comando `logfile` o implícitamente emitiendo algún otro comando. Si el usuario está en el modo de línea de comandos o en el modo por lotes y no especifica ningún nombre de archivo de registro, se usará el archivo predeterminado `/var/opt/perf/datafiles/logglob`. Si el usuario está en el modo interactivo y no especifica ningún nombre de archivo de registro, se le solicitará que proporcione uno o acepte el archivo predeterminado `/var/opt/perf/datafiles/logglob`.

Sintaxis

```
logfile [archivo de registro]
```

Instrucciones sobre su uso

Se puede especificar el nombre tanto del archivo de registro extraído como del archivo de registro sin formato. Si se especifica el nombre del archivo de registro extraído, toda la información será obtenida de este mismo archivo. No es necesario especificar ningún archivo de registro sin formato distinto al archivo de registro global `logglob`. Abrir `logglob` da acceso a todos los datos contenidos en los otros archivos de registro.

Los archivos de registro sin formato tienen los nombres siguientes:

<code>logglob</code>	archivo de registro global
<code>logappl</code>	archivo de registro de aplicación
<code>logproc</code>	archivo de registro de proceso
<code>logdev</code>	archivo de registro de dispositivo

logtran	archivo de registro de transacción
logls	archivo de registro de sistemas lógicos
logindx	archivo de registro de índice

Tras abrirse satisfactoriamente un archivo de registro, se imprimirá o mostrará un informe que muestra el contenido general del archivo de registro (o archivos de registro), como se ilustra en el ejemplo de la página siguiente.

```
Global      file: /var/opt/perf/datafiles/logglob version D
Application file: /var/opt/perf/datafiles/logappl
Process     file: /var/opt/perf/datafiles/logproc
Device      file: /var/opt/perf/datafiles/logdev
Transaction file: /var/opt/perf/datafiles/logtran
Index       file: /var/opt/perf/datafiles/logindx
System ID:  homer
System Type 9000/715 S/N 6667778899 O/S HP-UX B.10.20. A
Data Collector: SCOPE/UX C.02.30
File Created: 06/14/99
Data Covers: 27 days to 7/10/99
Shift is:    All Day
```

Data records available are:

```
Global Application Process Disk Volume Transaction
```

Maximum file sizes:

```
Global=10.0 Application=10.0 Process=20.0 Device=10.0 Transaction=10.0 MB
```

```
The first GLOBAL      record is on 06/14/99 at 12:00 AM
The first APPLICATION record is on 06/25/99 at 12:00 AM
The first PROCESS     record is on 07/06/99 at 12:01 AM
The first DEVICE      record is on 05/01/99 at 11:50 AM
The first TRANSACTION record is on 05/01/99 at 11:55 AM
The default starting date & time = 05/01/99 11:50 AM (FIRST + 0)
The default stopping date & time = 07/10/99 11:59 PM (LAST - 0)
```

El archivo de registro abierto podrá ser comprobado con el comando `show`, como se describe más adelante.

Puede abrirse otro archivo de registro en cualquier momento introduciendo otro comando `logfile`. Para que se pueda abrir el nuevo archivo de registro, todo archivo de registro que se encuentre abierto será cerrado.

Los comandos `resize` y `scan` requieren un archivo de registro para poder ser abiertos. Si no se encuentra ningún archivo de registro abierto, se ejecutará un comando implícito `logfile`.



No cambie el nombre de los archivos de registro sin formato. El acceso a estos archivos asume que se han especificado los nombres de archivo de registro estándares.

Si el usuario necesita tener más de un conjunto de archivos de registro en el mismo sistema, deberá crear un directorio aparte para cada conjunto de archivos. Aunque los nombres de los archivos de registro no pueden cambiarse, pueden usarse directorios diferentes. Si se desea cambiar el tamaño de los archivos de registro, deberá tenerse acceso de escritura/lectura a todos los archivos de registro.

menu

Use el comando `menu` para imprimir una lista de los comandos `utility` disponibles.

Sintaxis

menu

Ejemplo

```
utility> menu
Command Parameters      Function
HELP      topic]       Get information on commands and options
GUIDE
LOGFILE   [logname]     Specify a log file to be processed
LIST      [filename|*]  Specify the listing file
START     [startdate time] Set starting date & time for SCAN or ANALYZE
STOP      [stopdate time] Set ending date & time for SCAN or ANALYZE
DETAIL    [ON|OFF]      Set report detail for SCAN, PARMFILE, or ANALYZE
SHOW      [ALL]         Show the current program settings
PARMFILE  [parmfile]    Check parsing of a parameter file
SCAN      [logname]     Read the log file and produce a summary report
RESIZE    [GLOB|APPL|PROC|DEV|TRAN]
           [DAYS=][EMPTY=] Resize raw log files
CHECKDEF  [alarmdef]    Check parsing and set the alarmdef file
ANALYZE
! or Sh   [command]    Execute a system command
MENU or ?
EXIT or Q
utility>
```

parmfile

Use el comando `parmfile` para ver y comprobar la sintaxis de la configuración del archivo `parm` Componente Performance Collection que es usada para la recopilación de datos.

Sintaxis

parmfile [/directorypath/parmfile]

Instrucciones sobre su uso

El usuario puede usar el comando `parmfile` para hacer alguna de las siguientes acciones:

- Examinar el archivo `parm` para obtener advertencias de sintaxis y revisar la configuración resultante. Se comprobará la sintaxis de todos los parámetros y se notificarán los errores. Una vez finalice la comprobación de sintaxis, sólo se notificará la configuración aplicable.
- Averiguar la cantidad de espacio restante para definir aplicaciones.
- Si se ha especificado `detail on`, imprimir los contenidos especificados del archivo `parm` y toda configuración predeterminada que no haya sido invalidada, e imprimir las definiciones de aplicación.

En el modo por lotes, si no se ha especificado ningún archivo `parm`, se usará el archivo predeterminado `parm`.

En el modo interactivo, si no se ha especificado ningún archivo `parm`, se solicitará que se especifique uno.

Ejemplo

El comando `parmfile` comprueba la sintaxis del archivo `parm` actual y notifica toda advertencia o error. `Detail on` enuncia la configuración de parámetros de registro.

```
utility>
detail on
parmfile parm
```

Para realizar la tarea anterior usando los argumentos de la línea de comando, introduzca:

```
utility -xp -D
```

quit

El comando `quit` permite finalizar el programa `utility`. El comando `quit` equivale al comando `exit` del programa `utility`.

Sintaxis

```
quit
q
```

resize

Use el comando `resize` para administrar el espacio de su conjunto de archivos de registro sin formato. Este es el *único* programa que deberá usarse para cambiar de tamaño los archivos de registro sin formato con objeto de mantener la coordinación entre los archivos y sus estructuras de control internas. Si se usan otras herramientas se puede eliminar o destruir la validez de estas estructuras de control.

El programa `utility` *no puede* utilizarse para cambiar el tamaño de los archivos extraídos. Si se desea cambiar el tamaño de un archivo extraído, deberá usarse el programa `extract` para crear un nuevo archivo de registro extraído.

Sintaxis

```
resize [global      ] [days=maxdays] [empty=days] [yes  ]
      [application] [size=maxMB  ] [space=MB  ] [no   ]
      [process     ]                               [maybe]
      [device      ]
      [transaction]
```

Parámetros

<code>log file type</code>	Especifica el tipo de datos sin formato que el usuario desea cambiar de tamaño: <code>global</code> , de aplicación, proceso, dispositivo o transacción, que se corresponden con los archivos de registro sin formato <code>logglob</code> , <code>logappl</code> , <code>logproc</code> , <code>logdev</code> y <code>logtran</code> . Si el usuario no especifica un tipo de datos y está ejecutándose <code>utility</code> en el modo por lotes, el trabajo por lotes finalizará. Si el usuario está ejecutando <code>utility</code> de manera interactiva, se le solicitará que proporcione el tipo de datos basado en esos archivos de registro existentes.
<code>days & size</code>	Especifica el tamaño máximo del archivo de registro. El tamaño real dependerá de la cantidad de datos del archivo.
<code>empty & space</code>	Especifica la cantidad mínima de espacio requerida en el archivo tras finalizar la operación de cambio de tamaño. Este valor es usado para determinar si alguno de los datos que se encuentra en el archivo de registro debe ser eliminados en el proceso de cambio de tamaño.

Es de esperar que un archivo de registro no se llene hasta el número de días especificado tras una operación de cambio de tamaño. Esta función del comando `resize` puede usarse para minimizar el número de veces que un archivo de registro debe ser cambiado de tamaño por el recolector `scope` porque la acción de cambio de tamaño puede ocurrir en cualquier momento en que se llene el archivo. Usar `resize` para liberar cierta cantidad de espacio en un archivo de registro provocará que el archivo de registro sea cambiado de tamaño cuando el usuario lo determine.

Los valores `days` y `empty` serán introducidos en unidades de días; los valores `size` y `space` serán introducidos en unidades de megabytes. Los días son convertidos en megabytes usando un valor promedio de megabytes por día para el archivo de registro. Este factor de conversión variará dependiendo del tipo de datos que están siendo archivados y las características particulares del sistema en uso.

Para obtener factores de conversión promedio de megabytes por día más precisos, puede emitirse el comando `scan` del archivo de registro existente antes de emitirse el comando `resize`. El comando `scan` mide los índices de acumulación del sistema en uso. Si no se realiza ninguna exploración o si el factor de conversión medido resulta irreal, el comando `resize` usará el factor de conversión predeterminado para cada tipo de datos.

yes	Especifica que el cambio de tamaño debería realizarse incondicionalmente. Ésta es la acción predeterminada si <code>utility</code> no está siendo ejecutado de manera interactiva. Si no se especifica ninguna acción cuando <code>utility</code> esté ejecutándose de manera interactiva, se solicitará al usuario que proporcione una acción.
no	Especifica que el cambio de tamaño no debería realizarse. Este parámetro puede ser especificado como una acción si se desea ver el informe de cambio de tamaño pero no se desea realizar el cambio de tamaño en ese momento.
maybe	Especifica que <code>utility</code> debería decidir si cambiar el tamaño del archivo o no. Este parámetro obliga a <code>utility</code> a tomar una decisión basada en la cantidad de espacio libre existente en el archivo de registro (antes de ningún cambio de tamaño) y la cantidad de espacio especificada en el comando <code>resize</code> . Si el archivo de registro actual contiene como mínimo tanto espacio libre como se ha especificado, el cambio de tamaño <i>no</i> tendrá lugar. Si el archivo de registro actual contiene menos espacio libre del especificado, tendrá lugar el cambio de tamaño.
maybe (continuación)	Si el cambio de tamaño puede realizarse sin eliminar ningún dato del archivo de registro (por ejemplo, incrementando el tamaño máximo del archivo de registro o reduciendo el tamaño máximo del archivo de registro sin tener que eliminar ningún dato), el cambio de tamaño tendrá lugar. El parámetro <code>maybe</code> tiene como finalidad principal ser usado por ejecuciones por lote periódicas. Consulte la subsección “Ejemplos” que se muestra a continuación para obtener instrucciones sobre el uso del comando <code>resize</code> para dicha finalidad.

La tabla siguiente muestra parámetros de cambio de tamaño predeterminados.

Tabla 6 Parámetros de cambio de tamaño predeterminados

Parámetro	En caso de ejecución interactiva	En caso de ejecución por lotes
<code>log file type</code>	Se solicita al usuario cada tipo de archivo de registro disponible.	Sin valor predeterminado. Parámetro requerido.
<code>days size</code>	Tamaño de archivo actual.	Tamaño de archivo actual.
<code>empty space</code>	Cantidad de espacio libre actual o suficiente espacio libre para conservar todos los datos existentes en el archivo, lo que resulte ser más pequeño.	Cantidad de espacio libre actual o suficiente espacio libre para conservar todos los datos existentes en el archivo, lo que resulte ser más pequeño.
yes no maybe	El usuario recibirá un mensaje tras mostrarse los resultados de espacio de disco notificados.	Sí. El cambio de tamaño tendrá lugar.

Instrucciones sobre su uso

Antes de cambiar el tamaño de un archivo de registro, el usuario *deberá* impedir que Componente Performance Collection siga los pasos [Detención y reinicio de la recopilación de datos](#) en la página 40 del Capítulo 2.

Deberá abrirse un archivo de registro sin formato para que el cambio de tamaño tenga lugar. Abra el archivo de registro sin formato con el comando `logfile` antes de emitir el comando `resize`. Los archivos no pueden ser abiertos por ningún otro proceso.

El comando de cambio de tamaño crea el nuevo archivo `scopelog` en el directorio establecido por la variable de entorno `TMPDIR` antes de eliminar el archivo de registro original. Si la variable de entorno `TMPDIR` no se ha establecido, se usará entonces el directorio `/var/tmp (/tmp` en IBM AIX 4.1 y posterior) como ubicación temporal. El usuario deberá asegurarse de que hay suficiente espacio de disco en el directorio especificado por el `TMPDIR` o en el directorio `/var/tmp (/tmp` en IBM AIX 4.1 y posterior) para mantener el archivo de registro original antes de proceder con el proceso de cambio de tamaño.

Tras el cambio de tamaño, el archivo de datos constará de datos y espacio libre. Los datos retenidos se calculan como tamaño de archivo máximo menos espacio libre requerido. Se perderán todos los datos que sean eliminados durante la operación de cambio de tamaño. Si se desean guardar datos de archivo de registro durante periodos más largos, deberá usarse `extract` para copiar estos datos a un archivo extraído *antes* de proceder con la operación de cambio de tamaño.

Informes del comando `resize`

Al cambiar el tamaño de un archivo de registro sin formato se crea un informe estándar. Muestra las tres categorías interrelacionadas de espacio de disco, a saber: de tamaño de archivo máximo, de registros de datos y espacio libre, antes y después del proceso de cambio de tamaño. Por ejemplo:

```
resize global days=120;empty=10
empty space raised to match file size and data records
final resizing parameters:
file: logglob                                megabytes / day: 0.101199
---currently----                             --after resizing---
maximum size:  65 days (  6.6 mb)   120 days ( 12.1 mb)   83% increase
data records:  61 days (  6.2 mb)   61 days (  6.2 mb)   no data removed
empty space:    4 days (  0.5 mb)    59 days (  6.0 mb)  1225% increase
```

El valor megabytes por día permite realizar una conversión con días y megabytes. O bien es el valor obtenido durante la función de exploración, o un valor predeterminado correspondiente al tipo de datos que está siendo cambiado de tamaño.

La columna del extremo derecho contiene un resumen del cambio neto en cada categoría de espacio de archivo de registro. El tamaño y espacio máximos puede incrementarse, disminuir o permanecer invariable. Los registros de datos o bien no tienen datos eliminados o bien tienen una cantidad específica de datos eliminados durante el proceso de cambio de tamaño.

Si el proceso de cambio de tamaño se hace de manera interactiva y uno o más parámetros son predeterminados, el usuario podrá obtener un informe de cambio de tamaño preliminar. Este informe resume el contenido del archivo de registro actual y los parámetros que fueran proporcionados. Este informe se facilita para asistir en la respuesta a preguntas sobre parámetros no especificados. Por ejemplo:

```
resize global days=20
file resizing parameters (based on average daily
space estimates and user resizing parameters)
file: logglob                                megabytes / day: 0.101199
-----currently----- --after resizing--
maximum size: 65 days ( 6.6 mb)    20 days ( 2.0 mb)
data records: 61 days ( 6.2 mb)    ??
empty space:  4 days ( 0.5 mb)     ??
```

En el ejemplo anterior se solicita al usuario que proporcione la cantidad de espacio del archivo antes de ofrecerse un informe de cambio de tamaño final. Si no se proporciona ningún parámetro de acción para el cambio de tamaño interactivo, se preguntará al usuario si desea o no cambiar el tamaño del archivo de registro inmediatamente después del informe de cambio de tamaño final.

Ejemplos

Los comandos siguientes permiten cambiar el tamaño de un archivo de registro de proceso sin formato. La exploración tiene lugar antes del proceso de cambio de tamaño con objeto de incrementar la precisión de los cálculos de número de días.

```
logfile /var/opt/perf/datafiles/logglob
detail off
scan
resize process days=60 empty=30 yes
```

`days=60` especifica que se retendrá un máximo de 60 días de datos. `empty=30` especifica que 30 días de este archivo se encuentran libres en ese momento. Esto es, el archivo es cambiado de tamaño con no más de 30 días de datos en el archivo para dejar espacio para 30 días más de un total de 60 días de espacio. `yes` especifica que la operación de cambio de tamaño deberá tener lugar independientemente del espacio libre existente en ese momento.

El ejemplo siguiente muestra cómo puede usarse el comando `resize` en el modo por lotes para asegurarse de que los archivos de registro no se llenen durante la semana siguiente (obligando a `scope` a que los cambie de tamaño). El usuario podrá programar una secuencia de comandos cron mediante el comando `at` que especifica una cantidad de espacio mínima de, por ejemplo, 7 días, o incluso 10, para mayor seguridad.

La siguiente secuencia shell logra lo anterior:

```
echo detail off                                > utilin
echo scan                                      >> utilin
echo resize global      empty=10 maybe >> utilin
echo resize application empty=10 maybe >> utilin
echo resize process     empty=10 maybe >> utilin
echo resize device      empty=10 maybe >> utilin
echo quit                                                       >> utilin
utility < utilin > utilout 2> utilerr
```

Especificar `maybe` en lugar de `yes` impide que tenga lugar ninguna operación de cambio de tamaño si existen en ese momento 10 días o más de espacio libre en cualquiera de los archivos de registro. Adviértase que el tamaño de archivo máximo predeterminado se corresponde con el tamaño de archivo máximo correspondiente a cada archivo. Ello permite que el tamaño de los archivos sea cambiado por nuevos tamaños máximos sin que esta secuencia de comandos se vea afectada.

scan

El comando `scan` permite leer un archivo de registro y escribir un informe sobre su contenido. (Para obtener una descripción detallada del informe, consulte [Detalles del informe de exploración de utility](#) en la página 64 en el capítulo 3).

Sintaxis

`scan`

Instrucciones sobre su uso

El comando `scan` requiere un archivo de registro para poder ser abierto. El archivo de registro explorado es el primero de uno de los siguientes:

- El archivo de registro nombrado en el mismo comando `scan`.
- El último archivo de registro abierto por cualquier comando anterior.
- El archivo de registro predeterminado.

En este caso se solicitará a los usuarios interactivos que invaliden el nombre del archivo de registro predeterminado si así lo desean.

Los comandos siguientes afectan la operación de la función de exploración:

<code>detail</code>	Especifica la cantidad de detalles del informe. El predeterminado, <code>detail on</code> , especifica totalidad de detalles.
<code>list</code>	Redirecciona la salida a otro archivo. El predeterminado es incluir en dispositivo de lista estándar.
<code>start</code>	Especifica la fecha y hora del primer registro del archivo de registros que se desea explorar. El predeterminado es el inicio del archivo de registro.
<code>stop</code>	Especifica la fecha y hora del último registro del archivo de registros que se desea explorar. El predeterminado es el fin del archivo de registro.

Para más información sobre los comandos `detail`, `list`, `start` y `stop`, consulte las descripciones contenidas en este capítulo.

El informe del comando `scan` consta de 12 secciones. El usuario puede determinar qué secciones incluir en el informe emitiendo el comando `detail` antes de emitir `scan`.

Las siguientes cuatro secciones se imprimen siempre (incluso si se especifica `detail off`):

- Horas y fechas de inicio y parada reales de Scan
- Resumen de cobertura del recopilador
- Resumen de contenidos del archivo de registro
- Resumen de espacio libre del archivo de registro

Las secciones siguientes se imprimen si `detail on` (el predeterminado) ha sido especificado:

- Información global inicial del archivo `parm` e información de configuración del sistema
- Definiciones de aplicación iniciales del archivo `parm`
- Cambios globales del archivo `parm`
- Notificaciones de adición/eliminación de aplicación del archivo `parm`
- Notificaciones de tiempo de inactividad del recopilador
- Informes de resúmenes específicos de aplicación

La sección siguiente se imprime siempre que sean explorados datos de aplicación (incluso si se especifica `detail off`):

- Resumen general de aplicación

La sección siguiente se imprime siempre que sean explorados datos de proceso (incluso si se especifica `detail off`):

- Resumen de motivos de registro de procesos

Ejemplo

La exploración de un archivo de registro global predeterminado se inicia con registros que fueron registrados desde el 1 de junio de 1999, a las 7.00am, hasta la fecha y hora actuales.

```
utility>
logfile /var/opt/perf/datafiles/logglob
detail on
start 6/1/99 7:00 am
scan
```

Para realizar la tarea anterior usando los argumentos de la línea de comando, introduzca:

```
utility -D -b 6/1/99 7:00 am -xs
```

sh

`sh` permite introducir un comando shell sin salir de `utility` escribiendo `sh` o un signo de exclamación (!) seguido del comando shell.

Sintaxis

```
sh o! [shell command]
```

Parámetros

`sh ls` Ejecuta el comando `ls` y regresa a `utility`.

`!ls` Idem.

Instrucciones sobre su uso

Siguiendo la ejecución del comando en particular, se regresa a `utility`. Si el usuario desea emitir múltiples comandos shell sin regresar a `utility` después de cada uno de ellos, podrá iniciar un nuevo shell. Por ejemplo:

`sh ksh`

o bien

`!ksh`

show

El comando `show` enuncia los nombres de los archivos que están abiertos y el estado de los parámetros `utility` que pueden establecerse.

Sintaxis

`show [all]`

Ejemplos

`show` permite producir una lista que tiene un aspecto similar al siguiente:

```
Logfile: /var/opt/perf/datafiles/logglob
List: "stdout"
Detail: ON for ANALYZE, PARMFILE and SCAN functions
The default starting date & time = 10/08/99 08:17 AM (FIRST + 0)
The default stopping date & time = 11/20/99 11:59 PM (LAST - 0)
The default shift = 12:00 AM - 12:00 AM
```



La hora de turno predeterminado se muestra sólo a título informativo. La hora de turno no puede ser cambiada en `utility`.

show all permite producir una lista más detallada que tiene un aspecto similar al siguiente:

```
Logfile: /var/opt/perf/datafiles/logglob
Global      file: /var/opt/perf/datafiles/logglob
Application file: /var/opt/perf/datafiles/logappl
Process     file: /var/opt/perf/datafiles/logproc
Device      file: /var/opt/perf/datafiles/logdev
Transaction file: /var/opt/perf/datafiles/logtran
Index       file: /var/opt/perf/datafiles/logindx
System ID:  homer
System Type 9000/715 S/N 66677789  OS/ HP-UX B.10.20 A
Data Collector: SCOPE/UX C.02.30
File created: 10/08/99
Data Covers: 44 days to 11/20/99
Shift is:    All Day
Data records available are:
  Global Application Process Disk Volume Transaction
Maximum file sizes:
  Global=10.0 Application=10.0 Process=20.0 Device=10.0 Transaction 10.0
MB
List      "stdout"
Detail   ON for ANALYZE, PARMFILE and SCAN functions
The default starting date & time = 10/08/99 11:50 AM (FIRST + 0)
The default stopping date & time = 11/20/99 11:59 PM (LAST - 0)
The default shift = 12:00 AM - 12:00 AM
```

start

El comando `start` permite especificar el inicio del subconjunto de un archivo de registro que se desee explorar o analizar. `start` permite iniciar los procesos de `scan` o `analyze` para datos que fueron registrados en una fecha y hora determinadas.

La hora y fecha de inicio predeterminadas se establecen como la fecha y hora del primer registro, sea cual sea su tipo, en un archivo de registro abierto con el comando `logfile`.

Sintaxis

```
start      [fecha    [hora]]
            [today [-días]      [hora]]
            [last  [-días]      [hora]]
            [first [+días]      [hora]]
```

Parámetros

date	El formato de fecha dependerá del idioma original configurado en el sistema en uso. Si el usuario no usa ningún idioma original o ha establecido por defecto C para el idioma, el formato de fecha será <i>mm/dd/yy</i> (mes/día/año) o 06/30/99 en el caso del 30 de junio de 1999.
time	El formato de tiempo dependerá también del idioma original en uso. En el caso de C, el formato es <i>hh:mm am</i> o <i>hh:mm pm</i> (hora:minuto en formato de 12 horas con el sufijo am/pm) como por ejemplo 07:00 am en el caso de las 7 en punto de la mañana. Se acepta para todos los idiomas la especificación de hora en el formato de 24 horas. Por ejemplo, 23:30 serían las 11.30 pm. Si se introduce la fecha u hora en un formato inaceptable, se mostrará un ejemplo del formato correcto. Si no se proporciona hora de inicio, se propondrá la medianoche (12 am). La hora de inicio de medianoche de un día dado será el <i>comienzo</i> de ese día (00:00 en un formato de 24 horas).
today	Especifica el día actual. El parámetro <i>today-days</i> especifica el número de días con anterioridad a la fecha de hoy. Por ejemplo, <i>today-1</i> indica la fecha de ayer y <i>today-2</i> la fecha de antesdeayer.
last	Puede usarse para representar la última fecha contenida en el archivo de registro. El parámetro <i>last-days</i> especifica el número de días con anterioridad a la última fecha contenida en el archivo de registro.
first	Puede usarse para representar la primera fecha contenida en el archivo de registro. El parámetro <i>first+days</i> especifica el número de días tras la primera fecha contenida en el archivo de registro.

Instrucciones sobre su uso

El comando `start` es de utilidad en el caso de que se tenga un archivo de registro grande y no se desee explorar ni analizar el archivo en su totalidad. También puede usarse para especificar el periodo de tiempo determinado en el que se exploran los datos. Por ejemplo, se puede explorar un archivo de registro para datos que se registraron en un periodo que comenzó 14 días antes de la fecha actual especificando **today-14**.

Se puede usar el comando `stop` para limitar aún más los registros del archivo de registro que se desean explorar.

Si el usuario no sabe con seguridad si ha sido instalado en su sistema un idioma original determinado, podrá obligar a `utility` a usar el formato de fecha y hora C emitiendo la siguiente instrucción antes de ejecutar `utility`:

```
LANG=C; export LANG
```

o en C Shell

```
setenv LANG C
```

Ejemplo

La exploración de un archivo de registro global predeterminado se inicia con registros que fueron registrados desde el 05.08.99, a las 8.00 am, hasta la fecha y hora actuales.

```
utility>  
logfile /var/opt/perf/datafiles/logglob  
detail on  
start 8/5/99 8:00 AM  
scan
```

Para realizar la tarea anterior usando los argumentos de la línea de comando, introduzca:

```
utility -D -b 8/5/99 8:00 am -xs
```

stop

El comando `stop` permite especificar el final del subconjunto de un archivo de registro que se desea explorar o analizar. `Stop` permite finalizar el proceso de exploración o análisis para datos que fueron registrados en una hora y fecha determinadas.

La hora y fecha de parada predeterminadas se establecen como la fecha y hora del último registro, sea cual sea su tipo, en un archivo de registro abierto con el comando `logfile`.

Sintaxis

```
stop      [fecha      [hora]]
          [today    [-días]   [hora]]
          [last     [-días]   [hora]]
          [first    [+días]   [hora]]
```

Parámetros

<code>date</code>	El formato de fecha dependerá del idioma original configurado en el sistema en uso. Si el usuario no usa ningún idioma original o ha establecido por defecto <code>C</code> para el idioma, el formato de fecha será <code>mm/dd/yy</code> (mes/día/año) o <code>06/30/99</code> en el caso del 30 de junio de 1999.
<code>time</code>	El formato de tiempo dependerá también del idioma original en uso. En el caso de <code>C</code> , el formato es <code>hh:mm am</code> o <code>hh:mm pm</code> (hora:minuto en formato de 12 horas con el sufijo <code>am/pm</code>) como por ejemplo <code>07:00 am</code> en el caso de las 7 en punto de la mañana. Se acepta para todos los idiomas la especificación de hora en el formato de 24 horas. Por ejemplo, <code>23:30</code> serían las 11.30 pm. Si se introduce la fecha u hora en un formato inaceptable, se mostrará un ejemplo del formato correcto. Si no se proporciona hora de parada, se propondrá un minuto antes de la medianoche (<code>11:59 pm</code>). La hora de parada de medianoche (<code>12 am</code>) de un día dado será el <i>final</i> de ese día (<code>23:59</code> en un formato de 24 horas).
<code>today</code>	Especifica el día actual. El parámetro <code>today-days</code> especifica el número de días con anterioridad a la fecha de hoy. Por ejemplo, <code>today-1</code> indica la fecha de ayer y <code>today-2</code> la fecha de antesdeayer.
<code>last</code>	Puede usarse para representar la última fecha contenida en el archivo de registro. El parámetro <code>last-days</code> especifica el número de días con anterioridad a la última fecha contenida en el archivo de registro.
<code>first</code>	Puede usarse para representar la primera fecha contenida en el archivo de registro. El parámetro <code>first+days</code> especifica el número de días tras la primera fecha contenida en el archivo de registro.

Instrucciones sobre su uso

El comando `stop` es de utilidad en el caso de que se tenga un archivo de registro grande y no se desee explorar el archivo en su totalidad. También puede usarse para especificar el periodo de tiempo determinado en el que se exploran los datos. Por ejemplo, se puede explorar un archivo de registro de datos de un valor equivalente a siete días que fueron registrados empezando un mes antes de la fecha actual.

Si el usuario no sabe con seguridad si ha sido instalado en su sistema un idioma original determinado, podrá obligar a `utility` a usar el formato de fecha y hora C emitiendo la siguiente instrucción antes de ejecutar `utility`:

```
LANG=C; export LANG
```

o en C Shell

```
setenv LANG C
```

Ejemplo

La exploración de datos de un valor equivalente a 14 días empieza con registros registrados a partir del 5 de julio de 1999 a las 8:00 am y finaliza con el último registro registrado el 18 de julio de 1999 a las 11:59 pm.

```
utility>  
logfile /var/opt/perf/datafiles/logglob  
detail on  
start 7/5/99 8:00 am  
stop 7/18/99 11:59 pm  
scan
```

Para realizar la tarea anterior usando los argumentos de la línea de comando, introduzca:

```
utility -D -b 7/5/99 8:00 am -e 7/18/99 11:59pm -xs
```

6 Uso del programa `extract`

El programa `extract` presenta dos funciones principales: permite extraer datos de archivos de registro sin procesar y escribirlos en archivos de registro extraídos. `Extract` también permite exportar datos de archivos de registro que se utilizan en productos de análisis como hojas de cálculo.

➤ Tras la instalación inicial de Componente Performance Collection, se deben iniciar los servicios para que se complete la instalación de los archivos, antes de que funcione `extract`.

Las funciones `extract` y `export` *copian* los datos desde un archivo de registro y *no* se elimina ningún dato.

Componente Performance Collection utiliza tres tipos de archivos de registro:

- Los archivos de registro de `scope`, que contienen datos recopilados en Componente Performance Collection por el recopilador `scope`.
- Archivos de registro extraídos, que contienen datos extraídos a partir de archivos de registro de `scope` sin procesar.
- Archivos de registro DSI (integración de orígenes de datos), que contienen datos definidos por el usuario recopilados por orígenes externos, como aplicaciones y bases de datos. Los programas DSI de Componente Performance Collection registran posteriormente los datos.

El comando `extract` se utiliza para realizar las tareas siguientes:

- Extraer subconjuntos de datos de archivos de registro de `scope` sin procesar en un formato de archivo de registro extraído que sea adecuado para colocarlo en archivos, transportarlo entre sistemas y para que Performance Manager lo pueda analizar. *No* se pueden extraer datos a partir de archivos de registro DSI.
- Administrar datos de archivos de registro archivados mediante la extracción o exportación de datos a partir de archivos de formato extraídos, la anexión de datos a archivos de registro extraídos existentes y la creación de subconjuntos de datos por tipo, fecha y turno (hora del día).
- Exportar datos de archivos de registro de `scope` y de archivos de registro DSI sin procesar o extraídos a formatos ASCII, binario, archivo de datos o WK1 (hoja de cálculo) adecuados para generar informes y análisis o para importar a hojas de cálculo o paquetes de análisis similares.
- La función `extract` no puede generar datos resumidos. Los datos resumidos sólo puede generarlos la función `export`.
- Los límites máximos para las funciones `extract` y `export` son los siguientes:
 - El archivo de salida de la función `extract` no puede exceder de 3,5 GB.
 - El archivo de salida de la función `export` no puede exceder de 4 GB.

Para obtener ejemplos de cómo realizar diversas tareas y usar los comandos `extract`, consulte la ayuda en línea referente al programa `extract`.

Este capítulo aborda los temas siguientes:

- Ejecución del programa `extract`
- Uso del modo interactivo
- Interfaz de línea de comandos de extracción
- Introducción general de la función de exportación

Ejecución del programa `extract`

Existen tres formas de ejecutar el programa `extract`:

Modo de línea de comandos

El programa `extract` se controla mediante opciones de comandos y argumentos de la línea de comandos.

Modo interactivo

Se proporcionan parámetros y comandos interactivos al ejecutar el programa con `stdin` conectado a una terminal o estación de trabajo interactiva.

Los usuarios experimentados podrán especificar fácilmente sólo los comandos necesarios para realizar una tarea determinada. En caso de ser un usuario nuevo, se puede especificar el modo guiado para recibir más ayuda sobre cómo utilizar `extract`. En el modo guiado, se solicita al usuario que seleccione una opción de una lista para realizar una tarea. En el caso del modo guiado, los comandos interactivos que permiten realizar cada una de las tareas se muestran en una lista conforme son ejecutados, de manera que el usuario pueda ver cómo se usan. Es posible salir o volver a entrar en el modo guiado en cualquier momento.

Modo por lotes

Se puede ejecutar el programa y redirigir `stdin` a un archivo que contiene comandos y parámetros interactivos.

Sintaxis

La sintaxis correspondiente a la interfaz de línea de comandos es similar a la de las interfaces de línea de comandos UNIX estándar de otros programas y se describe en detalle en este capítulo.

En el caso del modo interactivo y por lotes, la sintaxis de comandos es la misma: un comando seguido de uno o más parámetros. Los comandos pueden introducirse en cualquier orden. Si el comando ha sido asociado a un parámetro en particular, dicho parámetro deberá ser introducido inmediatamente después del comando correspondiente.

Existen dos tipos de parámetros: los *requeridos* (para los cuales no existen opciones predeterminadas) y los *opcionales* (para los cuales se ofrecen opciones predeterminadas). El modo en que el programa `extract` gestiona estos parámetros dependerá del modo en el que se ejecute.

- En el modo interactivo, si falta un parámetro *opcional*, el programa mostrará el parámetro predeterminado y dejará que el usuario lo confirme o invalide. Si falta un parámetro *requerido*, el programa solicitará que se introduzca el parámetro.

- En el modo por lotes, si falta un parámetro *opcional*, el programa usará los valores predeterminados.
Si falta un parámetro *requerido*, el programa finalizará.

En el modo interactivo, los errores y datos que faltan se gestionarán de manera distinta que en el modo por lotes y en el modo de línea de comandos, porque se podrán proporcionar datos adicionales o corregir errores en el modo interactivo, pero no en el modo por lotes o en el modo de línea de comandos.

Uso del modo interactivo

Para utilizar el modo interactivo del programa `extract`, es necesario emitir una serie de comandos para ejecutar una tarea en particular.

Por ejemplo, si se desean exportar datos de aplicación recopilados a partir del 15 de mayo de 2003, desde el archivo de registro global predeterminado, se deben emitir los comandos siguientes después de invocar el programa `extract`

```
logfile /var/opt/perf/datafiles/logglob  
application detail  
start 5/15/2003  
export
```

El comando `logfile` abre `/var/opt/perf/datafiles/logglob`, el archivo de registro global predeterminado. El comando `start` especifica que sólo se exportarán los datos registrados después del 5/15/03. El comando `export` comienza a exportar los datos.

Interfaz de línea de comandos de extracción

Además de la sintaxis de comandos para el modo interactivo y el modo por lotes, las opciones de comando y sus argumentos pueden transferirse al programa `extract` mediante la interfaz de línea de comandos. La interfaz de línea de comandos funciona bien en entornos UNIX convencionales ya que permite que secuencias de comandos shell puedan invocar el programa `extract` con facilidad y que tanto las entradas como las salidas se redireccionen a canalizaciones UNIX.

Por ejemplo, el equivalente de la línea de comandos del ejemplo mostrado en la sección anterior [Uso del modo interactivo](#) en la página 97 es:

```
extract -l -a -b 5/15/02 -xp
```

En el modo de línea de comandos, el archivo de registro global `/var/opt/perf/datafiles/logglob` es el predeterminado y no es necesario especificarlo.

Los argumentos y opciones de línea de comandos se enuncian en la tabla siguiente. Las descripciones de comandos referenciados se encuentran en [Capítulo 7, Comandos del programa extract](#).

Tabla 7 Argumentos de línea de comandos

Opción de comando	Argumento		Descripción
-b	date	time	Especifica la fecha y la hora de inicio de una función <code>extract</code> o <code>export</code> . (Consulte el comando <code>start</code> en el capítulo 6).
-B		UNIX <i>hora de inicio</i>	Especifica la hora de inicio en formato UNIX de una función <code>extract</code> o <code>export</code> .
-e	date	time	Especifica la fecha y la hora de fin de una función <code>extract</code> o <code>export</code> . (Consulte el comando <code>stop</code> en el capítulo 6).
-E		UNIX <i>hora de fin</i>	Especifica la hora de fin en formato UNIX de una función <code>extract</code> o <code>export</code> .
-s	time-time	noweekends	Especifica la fecha de inicio y de fin para períodos específicos excluidos los fines de semana. (Consulte el comando <code>shift</code> en el capítulo 6).
-l	logfile		Especifica el archivo de registro de entrada. (Consulte el comando <code>logfile</code> en el capítulo 6). /var/opt/perf/datafiles/logglob es la opción predeterminada.
-r	export template file		Especifica un archivo de plantilla de exportación para la función <code>export</code> . (Consulte el comando <code>report</code> en el capítulo 6).
-C	classname	opt	Especifica datos de scope para <code>extract</code> o <code>export</code> , o datos autodescriptivos (DSI) para <code>export</code> . (Consulte el comando <code>class</code> en el capítulo 6). opt = detail (default) summary both off
-i			Especifica datos de scope para <code>extract</code> o datos de <code>export</code> desde sistemas lógicos.
-k			Exporta sólo procesos cerrados. Si se utiliza esta opción, se debe incluir la métrica <code>PROC_INTEREST</code> en <code>reptfile</code> .

Tabla 7 Argumentos de línea de comandos

Opción de comando	Argumento	Descripción
gapkcdzntuyiG ADZNTUYI		<p>Selecciona tipos de datos para <code>extract/</code> <code>export</code>:</p> <p><code>g</code> = detalles globales. (Consulte el comando global en el capítulo 6). La opción de detalles globales está desactivada de forma predeterminada.</p> <p><code>a</code> = detalles de aplicación. (Consulte el comando application en el capítulo 6).</p> <p><code>p</code> = detalles de proceso. (Consulte el comando process en el capítulo 6).</p> <p><code>k</code> = procesos cerrados. (Consulte el comando process en el capítulo 6).</p> <p><code>c</code> = detalles de configuración. (Consulte el comando configuration en el capítulo 6).</p> <p><code>d</code> = detalles de dispositivo de disco. (Consulte el comando disk en el capítulo 6).</p> <p><code>z</code> = detalles de lvolume. (Consulte el comando lvolume en el capítulo 6).</p> <p><code>n</code> = detalles de netif. (Consulte el comando netif en el capítulo 6).</p>

Tabla 7 Argumentos de línea de comandos

Opción de comando	Argumento	Descripción
gapkcdzntuyiG ADZNTUYI (continuación)		<p>t = detalles de transacción</p> <p>u = detalles de CPU</p> <p>y = detalles de sistema de archivos</p> <p>i = detalles de sistemas lógicos</p> <p>NOTA: las siguientes opciones de resumen son sólo para <code>export</code>; la función <code>extract</code> no admite el resumen de datos.</p> <p>G = resumen global El resumen global está desactivado de forma predeterminada.</p> <p>A = resumen de aplicación</p> <p>D = resumen de dispositivo de disco. (Consulte el comando disk en el capítulo 6).</p> <p>Z = resumen de <code>lvolume</code>. (Consulte el comando lvolume en el capítulo 6).</p> <p>N = resumen de <code>netif</code>. (Consulte el comando netif en el capítulo 6).</p> <p>I = resumen de sistemas lógicos</p>
gapkcdzntuyGA DZNTUY (continuación)		<p>T = resumen de transacción</p> <p>U = resumen de CPU. (Consulte el comando cpu en el capítulo 6).</p> <p>Y = resumen de sistema de archivos. (Consulte el comando filesystem en el capítulo 6).</p>
-v		Genera formatos de informes de salida detallada.
-f	filename ,new ,append ,purge	Envía datos de <code>extract</code> o <code>export</code> a un archivo. Si no se indica nombre de archivo, envía datos a archivos de salida predeterminados. (Consulte el comando output en el capítulo 6).
-ut		Muestra la fecha y la hora en formato UNIX en los datos de archivo de registro DSI exportados.

Tabla 7 Argumentos de línea de comandos

Opción de comando	Argumento	Descripción
-we	1.....7	Especifica días que se deben excluir de <code>export</code> ; 1=Lunes. (Consulte la descripción del comando <code>weekdays</code>).
-xp	xopt	Exporta datos a archivos de formato externo. (Consulte el comando <code>export</code> en el capítulo 6).
-xt	xopt	Extrae datos en un formato interno del sistema. (Consulte el comando <code>extract</code> en el capítulo 6). xopt = <i>dwmy</i> (Day Week Month Year) <i>dwmy</i> -[offset] <i>dwmy</i> [absolute]
-xw	week	Extrae datos de una semana natural. (Consulte el comando <code>weekly</code> en el capítulo 6).
-xm	month	Extrae datos de un mes natural. (Consulte el comando <code>monthly</code> en el capítulo 6).
-xy	year	Extrae datos de un año natural. (Consulte el comando <code>monthly</code> en el capítulo 6).
-? or ?		Muestra la sintaxis de la línea de comandos.

Al evaluar argumentos e introducir opciones de comando en la línea de comandos, se aplican las reglas siguientes:

- Los errores y datos que faltan se tratan exactamente igual que en el comando de modo por lotes correspondiente. Es decir, los datos que faltan tomarán los valores predeterminados si es posible y todos los errores provocarán el cierre inmediato del programa.
- La reproducción de comandos y resultados de comandos se deshabilita a menos que se utilice el argumento `-v` para habilitar el modo detallado.
- Si no se especifica una acción válida (`-xp`, `-xw`, `-xm`, `-xy` o `-xt`), `extract` inicia la lectura de comandos desde su archivo `stdin` una vez procesados todos los parámetros.
- Si se especifica una acción (`-xp`, `-xw`, `-xm`, `-xy` o `-xt`), el programa ejecutará estas opciones de comandos una vez evaluados todos los demás parámetros, con independencia de dónde estaban colocadas en la lista de parámetros.
- Si se especifica una acción en la línea de comandos, el programa `extract` no leerá el archivo `stdin`; en su lugar, finalizará la acción siguiente:

```
extract -f rxdata -r /var/opt/perf/rept1 -xp d-1 -G
```

Que se traduce en:

- f rxdata Coloca la salida en un archivo llamado rxdata del directorio actual
- r rept1 El archivo /var/opt/perf/rept1 contiene el formato export deseado.
- xp d-1 Exporta datos para este día menos 1 (ayer).
- G Exporta datos de resumen global.

Hay que tener en cuenta que la exportación real no se realiza hasta el final, por lo que el parámetro -G se procesa antes de realizar la exportación.

Asimismo, como no se ha especificado el archivo de registro, se utiliza el archivo predeterminado logglob.

Como se ha especificado una acción (-xp), una vez terminada la exportación, el programa extract finaliza sin leer el archivo stdin. Además, no se ha establecido el modo detallado con la opción de comando -v, por lo que se elimina toda salida externa a stdout.

Introducción general de la función de exportación

El comando export del programa extract convierte los datos de archivos de registro sin procesar, extraídos o DSI de Componente Performance Collection en archivos exportados. El comando export escribe archivos en uno de cuatro formatos posibles: ASCII, archivo de datos, binario y WK1 (hoja de cálculo). Los archivos exportados se pueden utilizar de diversas maneras, como informes, paquetes de gráficos personalizados, bases de datos y programas de análisis escritos por el usuario.

Cómo exportar datos

En su forma más sencilla, se pueden exportar datos:

- especificando el archivo de registro global predeterminado, /var/opt/perf/datafiles/logglob, desde el que se desean exportar datos,
- especificando el archivo de plantilla de exportación predeterminado, /var/opt/perf/reptfile, que define el formato de los datos exportados,
- iniciando la función de exportación.

Los datos exportados se colocan en el archivo de salida predeterminado llamado xfrdGLOBAL.asc en el directorio actual. El formato ASCII del archivo de salida es adecuado para la impresión.

Si se desea exportar algo que no sea este conjunto de datos predeterminados, se pueden utilizar otros comandos y archivos con el comando export.

Es posible exportar los siguientes tipos de datos:

<code>global</code>	Resúmenes de 5 minutos y por horas
<code>application</code>	Resúmenes de 5 minutos y por horas
<code>process</code>	Detalles de un minuto
<code>disk device</code>	Resúmenes de 5 minutos y por horas
<code>lvolume</code>	Resúmenes de 5 minutos y por horas
<code>transaction</code>	Resúmenes de 5 minutos y por horas
<code>configuration</code>	Un registro que contiene información del archivo <code>parm</code> e información de configuración del sistema, para cada vez que se inicia el recopilador de datos.
cualquier clase DSI	Intervalos y resúmenes para archivos de registro DSI
<code>netif</code>	Resúmenes de 5 minutos y por horas
<code>cpu</code>	Resúmenes de 5 minutos y por horas
<code>filesystem</code>	Resúmenes de 5 minutos y por horas

- Se puede especificar qué elementos de datos (métricas) son necesarios para cada tipo de datos.
- Se pueden indicar fechas de inicio y de fin para el período temporal en el que se recopilaron los datos junto con filtros de exclusión de turnos y de fines de semana.
- Se puede especificar el formato deseado para los datos exportados en un archivo de plantilla de exportación. Este archivo se puede crear mediante cualquier editor de texto o procesador de texto que permita guardar un archivo en formato ASCII (texto).
- Se puede utilizar también el archivo de plantilla de exportación predeterminado, `/var/opt/perf/reptfile`. Este archivo especifica las siguientes configuraciones de formatos de salida:
 - Formato de archivo ASCII
 - Un 0 (cero) para el valor que falta
 - Un espacio en blanco como separador de campos
 - Resúmenes de 60 minutos
 - Encabezados de columna incluidos
 - En la exportación se incluye un conjunto recomendado de métricas para un tipo de datos determinado



Si se extraen o exportan datos desde archivos de registro que se crean desde una plataforma específica, se recomienda utilizar el archivo `reptall` desde la misma plataforma. Esto se debe a que la lista de clases de métrica admitidas son diferentes en las distintas plataformas.

Tareas de exportación de ejemplo

Se proporcionan dos archivos de plantilla de exportación de ejemplo, `repthist` y `reptall`, con Componente Performance Collection. Estos archivos se encuentran en el directorio `/var/opt/perf/`. Se pueden utilizar `repthist` y `reptall` para realizar tareas de exportación comunes o como puntos de partida para tareas personalizadas, como la tarea descrita a continuación.

Generación de un informe de CPU imprimible

El archivo de plantilla de exportación `repthist` contiene las especificaciones para generar un gráfico de caracteres sobre el uso de CPU y de disco del sistema a lo largo del tiempo. Este gráfico consta de caracteres imprimibles a través de cualquier dispositivo capaz de imprimir 132 columnas. Por ejemplo, se pueden utilizar los siguientes comandos del programa `extract` para generar un gráfico de los últimos siete días y producir aproximadamente unas dos páginas (34 páginas si se especifican detalles de 5 minutos en lugar de resúmenes por horas).

```
logfile /var/opt/perf/datafiles/logglob
report /var/opt/perf/repthist
global summary
start today-7
export
```

Los datos exportados están en un archivo de exportación llamado `xfrsglobal.asc`. Para imprimirlo, escriba:

```
lp xfrsglobal.asc
```

Producción de un archivo de exportación personalizado

Si se desea crear un archivo de plantilla de exportación totalmente nuevo, se copia el archivo de plantilla de exportación y se personaliza mediante el comando `guide` del programa `extract`. En el modo guiado, se copia el archivo `reptall` del directorio `/var/opt/perf/` y se lee el archivo de registro de `scope` o DSI especificado para crear dinámicamente la lista de tipos de datos y nombres de métricas.

El archivo `reptall` contiene todas las métricas posibles para cada tipo de archivo de registro de `scope`, por lo que sólo es necesario quitar las marcas de comentario de aquellas métricas que sean de interés. Esto resulta más fácil que escribir de nuevo todo el archivo de plantilla de exportación.

Archivos de datos de exportación

Si se ha utilizado el comando `output` para especificar el nombre de un archivo de salida antes de emitir el comando `export`, todos los datos exportados se escribirán en este único archivo. Si se está ejecutando el programa `extract` de forma interactiva y se desean exportar datos directamente a la estación de trabajo (archivo de salida estándar), se debe especificar el comando `extract output stdout` antes de emitir el comando `export`.

Si el archivo de salida está establecido en la opción predeterminada, los datos exportados aparecen dispersos en un máximo de 14 archivos de salida predeterminados distintos en función del tipo de datos que se van a exportar.

Los nombres de los archivos de registro de exportación predeterminados son:

xfrdGLOBAL.ext	Archivo de datos de detalle globales
xfrsGLOBAL.ext	Archivo de datos de resumen global por horas
xfrdAPPLICATION.ext	Archivo de datos de detalle de aplicación
xfrsAPPLICATION.ext	Archivo de datos de resumen de aplicación por horas
xfrdPROCESS.ext	Archivo de datos de detalle de proceso
xfrdDISK.ext	Archivo de datos de detalle de dispositivo de disco
xfrsDISK.ext	Archivo de datos de resumen de dispositivo de disco por horas
xfrdVOLUME.ext	Archivo de datos de detalle del volumen lógico
xfrsVOLUME.ext	Archivo de datos de resumen de volumen lógico
xfrdNETIF.ext	Archivo de datos de detalle de NETIF
xfrsNETIF.ext	Archivo de datos de detalle de resumen de netif
xfrdCPU.ext	Archivos de datos de detalle de CPU
xfrsCPU.ext	Archivo de datos de resumen de CPU
xfrdFILESYSTEM.ext	Archivo de datos de detalle de sistema de archivos
xfrsFILESYSTEM.ext	Archivo de datos de resumen de sistema de archivos
xfrdTRANSACTION.ext	Archivo de datos de detalle de transacción
xfrsTRANSACTION.ext	Archivo de datos de resumen de transacción
xfrdCONFIGURATION.ext	Archivo de datos de configuración

donde ext= asc (ASCII), bin (binario), dat (archivo de datos) o wk1 (hoja de cálculo).



No se crea ningún archivo de salida *a menos que* se especifique el tipo y los elementos asociados que coinciden con los datos del archivo de plantilla de exportación antes de emitir el comando export.

Sintaxis del archivo de plantilla de exportación

El archivo de plantilla de exportación puede contener todo o parte de la información siguiente, en función del formato que se desee dar a los datos exportados y de lo que se quiera incluir en el archivo de exportación:

```
report      "export file title"
format      [ASCII]
            [datafile]
            [binary]
            [WK1] or
            [spreadsheet]
headings    [on]
            [off]
separator=  "char"
summary=value
missing=value
layout=single | multiple
output=filename
data type datatype
items
```

Parámetros

report	Especifica el título para el archivo de exportación. Para obtener más información, consulte la siguiente sección Título del archivo de exportación en la página 109.
formato	Especifica el formato para los datos exportados.

ASCII

El formato ASCII (o texto) es el más adecuado para copiar archivos en una impresora o terminal. No encierra campos entre comillas (").

Datafile (archivo de datos)

El formato `datafile` es similar al formato ASCII, excepto que los campos no numéricos se encierran entre comillas. Como las comillas impiden una estricta alineación de las columnas, no se recomienda usar archivos en formato `datafile` para la impresión directa. El formato `datafile` es el formato más fácil para importar a la mayoría de paquetes de hojas de cálculos y de gráficos.

Binary (binario)

El formato `binary` es más compacto porque los valores numéricos se representan como enteros binarios. Es el formato más adecuado para introducir información en programas de análisis escritos por el usuario porque es el que necesita menos conversión y mantiene la máxima precisión en las métricas. No es adecuado para la impresión directa.

WK1 (hoja de cálculo)

El formato `WK1` (hoja de cálculo) es compatible con Microsoft Excel y otros programas de hojas de cálculo y de gráficos.

headings	<p>Especifica si se incluyen o no los encabezados de columna para las métricas enumeradas en el archivo <code>export</code>. Si se especifica <code>headings off</code>, no se escriben encabezados de columna en el archivo. El primer registro del archivo son datos exportados. Si se especifica <code>headings on</code>, los formatos ASCII y <code>datafile</code> colocan el título de exportación y los encabezados de columna para cada columna de métricas escrita <i>antes</i> de los primeros registros de datos. Los encabezados de columna en archivos de formato <code>binary</code> contienen la descripción de las métricas del archivo. Los formatos <code>WK1</code> siempre contienen encabezados de columna.</p>
separator	<p>Especifica el carácter que se imprime entre cada campo en datos que tienen un formato ASCII o <code>datafile</code>. El carácter separador predeterminado es un espacio en blanco. Muchos programas prefieren una coma como separador de campos. Se puede especificar un separador como cualquier carácter imprimible o no imprimible.</p>
summary	<p>Especifica el número de minutos para cada intervalo de resumen. El valor determina cuánto tiempo se incluye en cada registro para los registros de resumen. El intervalo predeterminado es 60 minutos. El valor de resumen se puede establecer entre 5 y 1440 minutos (1 día).</p>
missing	<p>Especifica el valor de datos que se va a utilizar en lugar de los datos que faltan. El valor predeterminado para los datos que faltan es cero. Se puede especificar otro valor para diferenciar los datos que faltan de los datos que sean cero. Es posible que falte un elemento de dato si:</p> <ul style="list-style-type: none"> • no ha sido registrado por una versión determinada del recopilador de <code>scope</code> • no ha sido registrado por <code>scope</code> porque la instancia (<code>application</code>, <code>disk</code>, <code>transaction</code>, <code>netif</code>) a la que pertenece no estaba activa durante el intervalo o • en el caso de los archivos de registro DSI, no se han proporcionado datos al programa <code>dsilog</code> durante un intervalo, lo que ha generado “registros que faltan”. <p>Los registros que faltan se excluyen, de forma predeterminada, de los datos exportados.</p>
layout	<p>Especifica unos diseños <code>single</code> o <code>multiple</code> (salida por salida de registro) para tipos de datos como <code>application</code>, <code>disk</code>, <code>transaction</code>, <code>lvolume</code> o <code>netif</code>.</p> <ul style="list-style-type: none"> • El diseño único escribe una instancia por registro para cada aplicación que estaba activa durante el intervalo de tiempo. Ejemplo: un único disco exportado en un sólo registro. • El diseño múltiple escribe varias instancias en un único registro para cada intervalo de tiempo, reservando parte de ese registro para cada aplicación configurada. Ejemplo: todos los discos exportados en un sólo registro.

output	<p>Especifica el nombre del archivo de salida en el que se escribirán los datos exportados. Es posible especificar <code>output</code> para cada clase o <code>data type</code> exportado colocando <code>output nombre_de_archivo</code> justo después de la línea que indica el tipo de datos que inicia la lista de los elementos de datos exportados. Se puede especificar cualquier nombre de archivo válido para <code>output</code>.</p> <p>También se puede especificar el nombre de forma interactiva mediante el comando <code>output</code>. Cualquier nombre especificado invalida el nombre de archivo de salida predeterminado.</p>
data type	<p>Especifica uno de los tipos de datos exportables: <code>global</code>, <code>application</code>, <code>process</code>, <code>disk</code>, <code>transaction</code>, <code>lvolume</code>, <code>netif</code>, <code>configuration</code> o nombre de clase DSI. Esto inicia una sección del archivo de plantilla de exportación que enumera los elementos de datos que se van a copiar al exportar este tipo de datos.</p>
items	<p>Especifica las métricas que se van a incluir en el archivo exportado. Los nombres de métricas se enumeran, uno por línea, en el orden en el que se desea que aparezcan en el archivo resultante. Se debe especificar el tipo de datos adecuado antes de enumerar los elementos. El mismo archivo de plantilla de exportación puede incluir listas de elementos para todos los tipos de datos que se deseen. <i>Sólo</i> se hará referencia a un tipo de datos si se elige exportar dicho tipo.</p>

Se pueden utilizar los parámetros `output` y `layout` más de una vez dentro de un archivo de plantilla de exportación. Por ejemplo:

```
data type global
  output=myglobal
  gbl_cpu_total_util

data type application
  output=myapp
  layout=multiple
  app_cpu_total_util
```

Es posible tener más de un archivo de plantilla de exportación en el sistema. Cada uno de ellos puede definir un conjunto de formatos de archivos exportados que se adapte a una necesidad en particular. El comando `report` se usa para especificar el archivo de plantilla de exportación que se va a utilizar con la función `export`.



No se pueden especificar varios diseños distintos con un único tipo de datos. Por ejemplo, no es posible especificar **data type disk** una vez con **layout = multiple** y de nuevo con **layout = single** dentro del mismo archivo `export`.

Título del archivo de exportación

Se pueden sustituir los elementos siguientes en la cadena `export file title`:

<code>!date</code>	Fecha en la que se ejecutó la función <code>export</code> .
<code>!time</code>	Hora en la que se ejecutó la función <code>export</code> .
<code>!logfile</code>	Nombre completo del archivo de registro de origen.
<code>!class</code>	Tipo de datos solicitado.
<code>!collector</code>	Nombre y versión del programa recopilador (no es válido con archivos de registro DSI).
<code>!system_id</code>	Identificador del sistema que ha recopilado los datos (no es válido con archivos de registro DSI).

Por ejemplo, la cadena

```
report "!system_id data from !logfile on !date !time"
```

genera un título del archivo de exportación similar a

```
barkley data from logglob on 02/02/99 08:30 AM
```

Creación de un gráfico o informe personalizado

Para crear un gráfico o informe personalizado que contenga datos globales y de aplicación exportados, se hace lo siguiente:

- 1 Determinar qué elementos de datos (métricas) se necesitan para cada tipo de datos y en qué formato se tendrá acceso a ellos.

Para este ejemplo, se desea tener un archivo ASCII sin encabezados y con campos separados por comas.

- 2 Crear y guardar el siguiente archivo de plantilla de exportación ASCII en el directorio `/var/opt/perf/`. El archivo se llamará `report1`.

```
REPORT "sample export template file (report1)"  
FORMAT ASCII  
HEADINGS OFF
```

```
DATA TYPE GLOBAL  
  GBL_CPU_TOTAL_UTIL  
  GBL_DISK_PHYS_IO_RATE
```

```
DATA TYPE APPLICATION  
  APP_CPU_TOTAL_UTIL  
  APP_DISK_PHYS_IO_RATE  
  APP_ALIVE_PROCESSES
```

- 3 Ejecutar el programa `extract`.
- 4 Emitir el comando `report` para especificar el archivo de plantilla de exportación creado.
report /var/opt/perf/report1

- 5 Especificar los datos de resúmenes globales y los datos de resúmenes de aplicaciones mediante los comandos `global` y `application`.

```
global summary
application summary
```

- 6 Emitir el comando `export` para iniciar la exportación.

```
export
```

- 7 Como no se ha especificado donde debería obtener el programa los datos de rendimiento, se solicitará esta información. En este ejemplo, como el archivo de registro predeterminado es correcto, sólo se debe pulsar **Entrar**.

- 8 La salida debe tener un aspecto similar al siguiente:

```
exporting global data .....50%.....100%
exporting application data .....50%.....100%
The exported file contains 31 days of data from 01/01/99 to 01/31/99

              examined  exported
data type           records  records    space
-----
global      summaries                736  0.20 Mb
application summaries            2560  0.71 Mb
-----
                                0.91 Mb
```

Los dos archivos que se acaban de crear, `xfrsGLOBAL.asc` y `xfrsAPPLICATION.asc`, contienen los datos de resúmenes globales y de aplicaciones en el formato especificado.

Salida de datos exportados

El contenido de cada archivo exportado es el siguiente:

línea de título de exportación Si se han especificado `export title` y `headings on`.

Nombres (`application`, `netif`, `lvolume` o `transaction`) Si se ha especificado `headings on` junto con un archivo de diseño múltiple.

Línea de encabezado 1 Si se ha especificado `headings on`.

Línea de encabezado 2 Si se ha especificado `headings on`.

primer registro de datos

segundo registro de datos

...

último registro de datos

Las líneas de título y de encabezado del informe no se repiten en el archivo.

Notas sobre los formatos ASCII y datafile

Los datos de estos archivos de formato están en un formato ASCII imprimible. Los formatos ASCII y datafile son idénticos excepto que en éste último, todos los campos no numéricos están entre comillas. Incluso la información de encabezado datafile está entre comillas.

El formato de archivo ASCII no encierra campos entre comillas. Por lo tanto, los datos de archivos ASCII aparecerán correctamente alineados una vez impresos.

Los valores numéricos tienen un formato basado en el rango y la precisión interna. Como todos los campos no tendrán la misma longitud, es preciso especificar el separador que se va a utilizar para el inicio de cada campo.

El carácter separador especificado por el usuario (o el espacio en blanco predeterminado) separa los campos independientes en formatos ASCII y `datafile`. Los espacios en blanco, que se utilizan como separadores, pueden resultar más atractivos desde el punto de vista visual a la hora de imprimir el informe. Otros caracteres pueden resultar más útiles como separadores para leer el archivo de plantilla de exportación con otro programa.

El uso de la coma como separador se admite en muchas aplicaciones pero algunos elementos de datos pueden contener comas *que no sean separadores*. Estas comas pueden confundir a los programas de análisis. Los formatos de fecha y hora pueden contener diversos caracteres especiales basados en el idioma nativo especificado al ejecutar el programa `extract`.



Para utilizar un carácter especial no imprimible como separador, hay que introducirlo en el archivo de plantilla de exportación justo después de la primera comilla en el parámetro `separator`.



- La mayoría de las hojas de cálculo aceptan archivos en formato `datafile` si se utiliza `separator=","`.
- Muchos paquetes de hojas de cálculo aceptan un máximo de 256 columnas en una única hoja. Hay que tener cuidado al exportar tipos de diseño múltiple de datos porque es fácil generar más de 256 elementos en total. Se puede utilizar la salida del comando `report archivo_de_informe,show` para determinar si es probable que se vea este problema.
- Si se dispone de una impresora que admite el subrayado, se puede crear una copia impresa más atractiva especificando un formato ASCII y el carácter de barra vertical (`separator=|`) e imprimiendo luego el archivo con el subrayado activado.

Notas sobre el formato binario

En archivos de formato `binary`, los valores numéricos se escriben en enteros de 32 bits. Esto puede ahorrar espacio al reducir el tamaño del archivo global pero el programa debe ser capaz de leer archivos `binary`. No se recomienda copiar un archivo de formato `binary` en una impresora o terminal.

En formato `binary`, los datos no numéricos se escriben igual que en formato ASCII excepto que no se utilizan caracteres separadores. Para utilizar un archivo de formato `binary`, se debe usar el informe de diseño de registro impreso por `extract` al especificar **report archivo_de_informe, show**. Este informe proporciona el byte de inicio para cada elemento especificado.

Para mantener la máxima precisión y evitar representaciones de punto flotante `binary` no estándar, todos los valores numéricos se escriben como enteros de 32 bits a escala. Algunos elementos se pueden multiplicar por una constante antes de que se trunquen en un formato de entero.

Por ejemplo, el número de segundos que se utilizó la CPU se multiplica por 1000 antes de truncarse. Para convertir el valor del archivo exportado de nuevo en el número real de segundos, hay que dividirlo por 1000. A fin de facilitar la conversión, se debe especificar **headings on** para escribir los factores de escala en el archivo exportado. Los registros de título y encabezado especial del informe se escriben en archivos de formato `binary`, lo que ayuda a la interpretación de programación.

Los enteros binarios se escriben en el formato que es nativo del sistema en el que se ejecuta el programa `extract`. Por ejemplo, los sistemas Intel escriben enteros “little endian” mientras que los sistemas HP-UX, IBM AIX y Sun escriben enteros “big endian”. Hay que tener cuidado al transportar archivos exportados binarios a sistemas que utilizan “endians” diferentes.

Diseño de registros de encabezado binarios

Cada registro de un archivo exportado en formato `binary` contiene un encabezado de registro de 16 bytes especial que precede a los datos especificados por el usuario. El comando `report archivo_de_informe, show` incluye los cuatro campos siguientes que constituyen este encabezado de registro:

Métricas de encabezados de registros binarios

<code>Record Length</code>	Número de bytes en el registro, incluido el encabezado de registro de 16 bytes.
<code>Record ID</code>	Número que identifica el tipo de registro (ver a continuación).
<code>Date_Seconds</code>	Hora desde el 1 de enero de 1970 (en segundos).
<code>Number_of_vars</code>	Número de entradas de repetición en este registro.

La métrica `Record ID` identifica exclusivamente el tipo de datos contenidos en el registro. Los valores actuales de `Record ID` son:

- 1 Registro de título
- 2 Primer registro de encabezado (contiene números de elementos)
- 3 Segundo registro de encabezado (contiene factores de escala de elementos)
- 4 Registro de nombre de aplicación (para archivos de aplicación de varias instancias)
- 5 Registro de nombre de transacción (para archivos de transacción de varias instancias)
- 7 Registro de nombre de dispositivo de disco (para archivos de dispositivo de disco de varias instancias)
- 8 Registro de nombre de volumen lógico (para archivos de volumen lógico de varias instancias)
- 9 Registro de nombre de netif (para archivos de netif de varias instancias)
- 10 Registro de nombre de sistema de archivos (para archivos de netif de varias instancias)
- 11 Registro de nombre de CPU (para archivos de netif de varias instancias)


```

1 Global Data Record ( 5 minute detail record)
101 Global Data Record (60 minute summary record)
2 Application Data Record ( 5 minute detail record)
102 Application Data Record (60 minute summary record)
3 Process Data Record ( 1 minute detail record)
4 Configuration Data Record
7 Disk Device Data Record ( 5 minute detail record)
107 Disk Device Data Record (60 minute summary record)
8 Logical Volume Data Record ( 5 minute detail record)
108 Logical Volume Data Record (60 minute summary record)
9 Filesystem Data Record ( 5 minute detail record)
109 Filesystem Data Record (60 minute summary record)
11 Netif Data Record ( 5 minute detail record)
111 Netif Data Record (60 minute summary record)
12 Transaction Data Record ( 5 minute detail record)
112 Transaction Data Record (60 minute summary record)
13 CPU Data Record ( 5 minute detail record)
113 CPU Data Record ( 60 minute summary record)
ClassID +1,000,000 Class Data Record ( 5 minute detail record)
ClassID +1,000,000+100 Class Data Record (60 minute summary record)

```

La métrica `Date_Seconds` es idéntica a la métrica `Date_Seconds` seleccionable por el usuario y se proporciona para que se puedan explorar los registros con facilidad al buscar las fechas y horas que se deseen.

La métrica `Number_of_vars` indica cuántos grupos de campos de repetición están incluidos en el registro. Para tipos de datos de una única instancia, este valor es cero.

Para registros de aplicaciones de varias instancias, la métrica `Number_of_vars` es el número de aplicaciones configuradas. Para registros de dispositivos de disco de varias instancias, la métrica `Number_of_vars` es el número de dispositivos de disco configurados. Para todos los registros de encabezado, esta métrica es el número máximo de grupos de repetición permitido.

Los archivos de formato `binary` tienen formatos especiales para los registros de título y encabezado. Estos registros contienen la información necesaria para describir el contenido del archivo de modo que un programa pueda interpretarlo correctamente. Si se especifica `headings off`, sólo los registros de datos estarán en el archivo. Si se especifica `headings on`, los siguientes registros precederán a todos los registros de datos.

Registros de encabezado binarios

Registro de título	Este registro (Record ID -1) se escribe siempre que los encabezados estén activados, independientemente de si el usuario ha especificado un título de registro. Contiene información sobre el archivo de registro y su origen.
Primer registro de encabezado	El primer registro de encabezado (Record ID -2) contiene una lista de números de identificación de elementos exclusivos correspondientes a los elementos incluidos en el archivo de registro. La posición de los números de ID de elementos se puede utilizar para determinar la posición y el tamaño de cada elemento exportado en el archivo.
Segundo registro de encabezado	El segundo registro de encabezado (Record ID -3) contiene una lista de factores de escala correspondientes a los elementos exportados. Para obtener información más detallada, consulte “Factores de escala” posteriormente en esta sección.
Registro de nombre de aplicación	Este registro (Record ID -4) sólo estará presente en los archivos de datos de aplicación que utilizan el formato de diseño múltiple. Enumera los nombres de las aplicaciones que corresponden a los grupos de métricas de aplicación en el resto del archivo.
Registro de nombre de transacción	Este registro (Record ID -5) sólo estará presente en los archivos de datos de seguimiento de transacciones que utilizan el formato de diseño múltiple. Enumera los nombres de las transacciones que corresponden a los grupos de métricas de transacción en el resto del archivo.
Registro de nombre de dispositivo de disco	Este registro (Record ID -7) sólo estará presente en los archivos de datos de dispositivo de disco que utilizan el formato de diseño múltiple. Enumera los nombres de dispositivos de disco que corresponden a los grupos de métricas de dispositivo de disco en el resto del archivo.
Registro de nombre de volumen lógico	Este registro (Record ID -8) sólo estará presente en los archivos de datos de volumen lógico que utilizan el formato de diseño múltiple. Enumera los nombres de volúmenes lógicos que corresponden a los grupos de métricas de volumen lógico en el resto del archivo.
Registro de nombre de netif	Este registro (Record ID -9) sólo estará presente en los archivos de datos (LAN) de netif que utilizan el formato de diseño múltiple. Enumera los nombres de dispositivos de netif que corresponden a los grupos de métricas de dispositivo de netif en el resto del archivo.
Registro de nombre de sistema de archivos	Este registro (Record ID -12) sólo estará presente en los archivos de datos de sistema de archivos que utilizan el formato de diseño múltiple. Enumera los nombres de los sistemas de archivos que corresponden a los grupos de métricas de sistema de archivos en el resto del archivo.
Registro de nombre de CPU	Este registro (Record ID -13) sólo estará presente en los archivos de datos de CPU que utilizan el formato de diseño múltiple. Enumera los nombres de CPU que corresponden a los grupos de métricas de CPU en el resto del archivo.

Registro de título binario

El registro de título para archivos BINARY contiene información diseñada que ayuda a la interpretación de programación del contenido del archivo exportado. Este registro se escribirá en el archivo exportado siempre que se haya especificado `headings on`.

El contenido del registro del título binario es el siguiente:

Record Length	4 byte Int	Length of Title Record
Record ID	4 byte Int	-1
Date_Seconds	4 byte Int	Date exported file was created
Number_of_vars	4 byte Int	Maximum number of repeating variables
Size of Fixed Area	4 byte Int	Bytes in nonvariable part of record
Size of Variable Entry	4 byte Int	Bytes in each variable entry
GMT Time Offset	4 byte Int	Seconds offset from Greenwich Mean Time
Daylight Savings Time	4 byte Int	=1 indicates Daylight Savings Time
System ID	40 Characters,	System Identification
Collector Version	16 Characters,	Name & version of the data collector
Log File Name	72 Characters,	Name of the source log file
Report Title	100 Characters,	User specified report title

Las métricas `Date_Seconds`, `GMT Time Offset` y `Daylight Savings Time` en el registro del título binario se aplican al sistema y a la hora en los que se creó el archivo `export`. Si no es el mismo sistema que registró los datos, estos campos no pueden reflejar correctamente los datos del archivo.

Registro de identificación de elementos binario

El primer registro de encabezado (Record ID -2) del archivo binario contiene los números de elementos exclusivos para cada elemento exportado. Cada ID de elemento es un número entero de 4 bytes que puede tener referencias cruzadas mediante el archivo `rxitemid` proporcionado con este producto. Los campos de ID de elemento se alinean con los campos de datos que representan en el resto del archivo. Todos los elementos de datos exportados binarios ocuparán un múltiplo de 4 bytes en el archivo exportado y cada uno comenzará en un espacio limitado a 4 bytes. Si un elemento de dato requiere más 4 bytes de espacio, su campo de ID de elemento correspondiente se completará con ceros a la izquierda.

Por ejemplo, la métrica de proceso `Program` requiere 16 bytes. Sus registros de datos y de ID de elementos serían:

```
Header 1 (Item Id Record) ...| 0| 0| 0|12012|
Process Data Record      |Prog|ram_|name| _aaa|
```

Registro de factor de escala binario

El segundo registro de encabezado (Record ID -3) del archivo binario contiene los factores de escala para cada uno de los elementos exportados. Los elementos numéricos se exportan a archivos binarios como enteros de 32 bits (4 bytes) para minimizar los problemas con el modo en que distintas arquitecturas de ordenadores implementan el punto flotante. Antes de truncarse para que quepan en el formato de entero, muchos elementos se multiplican por un factor de escala fijo. Esto permite que los números de punto flotante se expresen como fracción, utilizando el factor de escala como denominador.

Cada factor de escala es un entero de 32 bits (4 bytes) que coincide con la mayoría de elementos de datos. Se utilizan valores especiales para los factores de escala, que indican métricas no numéricas y otras con valores especiales.

Factores de escala especiales

Las métricas no numéricas, como campos ASCII, tienen unos factores de escala de cero. No debería producirse un factor de escala 1 negativo pero si sucede, indica un error interno en el programa de extracción y se debe informar sobre ello.

El formato DATE es el formato MPE CALENDAR en los 16 bits menos significativos del campo (los 16 bits más a la derecha). El factor de escala de la fecha es 512. Al ajustarlo como un entero de 32 bits (dividiendo por 512), se aísla el año como parte del entero de la fecha y el día del año (dividido por 512) como parte fraccionaria.

TIME es un campo binario de 4 bytes (hora, minuto, segundo, décimas de segundo). El factor de escala para la hora es 65536. Al dividir por 65536, se forma un número en el que la parte del entero es la (hora * 256) + minuto.

Es más fácil manejar el valor `Date_Seconds` en un archivo binario.

Registro de nombre de aplicación

Al exportar datos de aplicación en formato de diseño múltiple, se escribe un registro de nombre de aplicación especial para identificar los grupos de aplicaciones. Para archivos de formato binario, este registro tiene un ID de registro -4. Consta del encabezado de 16 bytes del registro binario y un nombre de aplicación de 20 bytes para cada aplicación que se haya definido en la fecha de inicio de los datos exportados.

Si se agregan o eliminan aplicaciones durante el tiempo indicado en la extracción de datos, el registro de nombre de aplicación se repite con los nuevos nombres de aplicación.

Registro de nombre de transacción

Al exportar datos de transacción en formato de diseño múltiple, se escribe un registro de nombre de transacción especial para identificar el nombre de transacción de aplicación. Para archivos de formato binario, este registro tiene un ID de registro -5. Consta del encabezado de 16 bytes del registro binario y un nombre de transacción de aplicación truncado de 60 bytes para cada transacción que se haya configurado en la fecha de inicio de los datos exportados. Si se agregan transacciones durante el tiempo indicado en la extracción de datos, el registro de nombre de transacción se repetirá con el nuevo nombre de transacción de aplicación anexo al final de la lista original. Las transacciones que se eliminan tras el inicio de la extracción de datos no se suprimirán del registro de datos de diseño múltiple.

Registro de nombre de dispositivo de disco

Al exportar datos de dispositivo de disco en formato de diseño múltiple, se escribe un registro de nombre de dispositivo de disco especial para identificar el nombre del dispositivo de disco. Para archivos de formato binario, este registro tiene un ID de registro -7. Consta del encabezado de 16 bytes del registro binario y un nombre de dispositivo de disco de 20 bytes para cada dispositivo de disco que se haya configurado en la fecha de inicio de los datos exportados.

Si se agregan dispositivos de disco durante el tiempo indicado en la extracción de datos, el registro de nombre de dispositivo de disco se repetirá con el nuevo nombre de dispositivo de disco anexado al final de la lista original. Los dispositivos de disco que se eliminan tras el inicio de la extracción de datos no se suprimirán del registro de datos de diseño múltiple.

Registro de nombre de volumen lógico

Al exportar datos de volumen lógico en formato de diseño múltiple, se escribe un registro de nombre de volumen lógico especial para identificar el nombre del volumen lógico. Para archivos de formato binario, este registro tiene un ID de registro -8. Consta del encabezado de 16 bytes del registro binario y un nombre de dispositivo de disco de 20 bytes para cada volumen lógico que se haya configurado en la fecha de inicio de los datos exportados.

Si se agregan volúmenes lógicos durante el tiempo indicado en la extracción de datos, el registro de nombre de volumen lógico se repetirá con el nuevo nombre de volumen lógico anexado al final de la lista original. Los volúmenes lógicos que se eliminan tras el inicio de la extracción de datos no se suprimirán del registro de datos de diseño múltiple.

Registro de nombre de netif

Al exportar datos de `netif` en formato de diseño múltiple, se escribe un registro de nombre de `netif` especial para identificar el nombre del dispositivo de `netif`. Para archivos de formato binario, este registro tiene un ID de registro -11. Consta del encabezado de 16 bytes del registro binario y un nombre de dispositivo de `netif` de 20 bytes para cada dispositivo de `netif` que se haya configurado en la fecha de inicio de los datos exportados.

Si se agregan dispositivos de `netif` durante el tiempo indicado en la extracción de datos, el registro de nombre de `netif` se repetirá con el nuevo nombre de dispositivo anexado al final de la lista original. Los dispositivos de `netif` que se eliminan tras el inicio de la extracción de datos no se suprimirán del registro de datos de diseño múltiple.

7 Comandos del programa extract

En el presente capítulo se describen los comandos del programa `extract`. Incluye una tabla en la que se muestra la sintaxis del comando, una tabla de comandos para extraer y exportar datos y una sección de referencia de comandos en la que se describen los comandos en orden alfabético

Los comandos y parámetros de `extract` pueden introducirse con cualquier combinación de minúsculas y mayúsculas. Sólo son necesarias las tres primeras letras del nombre del comando, *excepto* para los comandos `weekdays` y `weekly`, para los que es necesario introducir todo el nombre. Por ejemplo, el comando `application detail` puede abreviarse como `app det`.

Para obtener ejemplos de cómo usar estos comandos consulte la ayuda en línea referente al programa `extract`.

En la tabla que se muestra en las páginas siguientes se resume la sintaxis de los comandos de `extract` y sus parámetros.



La función `extract` no puede generar datos resumidos. Los datos resumidos sólo puede generarlos la función `export`.

Tabla 8 Comandos de extract: Sintaxis y parámetros

Comando	Parámetro
<code>application</code>	<code>on</code> <code>detail</code> <code>summary</code> (sólo export) <code>both</code> (sólo export) <code>off</code> (predeterminado)
<code>class</code>	<code>detail</code> (predeterminado) <code>summary</code> (sólo export) <code>both</code> (sólo export) <code>off</code>
<code>cpu</code>	<code>detail</code> <code>summary</code> (sólo export) <code>both</code> (sólo export) <code>off</code> (predeterminado)
<code>configuration</code>	<code>on</code> <code>detail</code> <code>off</code> (predeterminado)

Tabla 8 Comandos de extract: Sintaxis y parámetros

Comando	Parámetro
disk	on detail summary (sólo export) both (sólo export) off (predeterminado)
exit	
e	
export	day[ddd] [-días] week [ss] [-semanas] month[mm] [-meses] year [aa] [-años]
extract	day[ddd] [-días] week [ss] [-semanas] month[mm] [-meses] year [aa] [-años]
filesystem	detail summary (sólo export) both (sólo export) off (predeterminado)
global	on detail (predeterminado) summary (sólo export) both (sólo export) off
guide	
help	
list	filename *
logfile	logfile
lvolume	on detail summary (sólo export) both (sólo export) off (predeterminado)
menu	
monthl	
y	aaamm mm

Tabla 8 Comandos de extract: Sintaxis y parámetros

Comando	Parámetro
netif	on detail summary (sólo export) both (sólo export) off (predeterminado)
output	<i>archivosalida</i> ,new ,purgeboth ,append
process	on detail [app#[-#],...] off (predeterminado) killed
quit q	
report	[<i>archivo plantilla exportación</i>], show
shift	<i>horainicio- horaparada</i> all day noweekends
sh !	shell command
show	all
start	<i>fecha [hora]</i> today[- <i>días</i>][<i>hora</i>] last[- <i>días</i>][<i>hora</i>] first[+ <i>días</i>][<i>hora</i>]
stop	<i>date [hora]</i> today[- <i>días</i>][<i>hora</i>] last[- <i>días</i>][<i>hora</i>] first[+ <i>días</i>][<i>hora</i>]
transaction	on detail summary (sólo export) both (sólo export) off (predeterminado)
weekdays	1.....7
weekly	<i>aass</i> <i>ss</i>
yearly	<i>aaaa</i> <i>aa</i>

En la siguiente tabla se enumeran los comandos usados para extraer y exportar datos y los tipos de archivos de registro usados (archivos de registro scope o archivos de registro DSI).

Tabla 9 Comandos de extract: extracción y exportación de datos

Comando	Extraer datos	Exportar datos	Archivos registro scope	Archivos registro DSI
application	x	x	x	
class	x	x	x	x
configuration		x	x	
cpu	x	x	x	
disk	x	x	x	
export		x	x	x
extract	x		x	
filesystem	x	x	x	
global	x	x	x	
logfile	x	x	x	x
lvolume	x	x	x	
monthly	x		x	
netif		x	x	
output	x	x	x	x
process	x	x	x	
report		x	x	x
shift	x		x	x
start	x	x	x	x
stop	x	x	x	x
transaction	x	x	x	x
weekdays		x	x	x
weekly	x		x	
yearly	x		x	
logicalsystems	x	x	x	

application

Use el comando `application` para especificar el tipo de datos de aplicación que se extraerán o exportarán.

El valor predeterminado es `application off`

Sintaxis

```
application [on]
            [detail]
            [summary]
            [both]
            [off]
```

Parámetros

<code>on</code> o <code>detail</code>	Especifica que los datos sin procesar, con los detalles de cinco minutos, deben extraerse o exportarse.
<code>summary</code> (sólo <code>export</code>)	Especifica que los datos deben resumirse por: <ul style="list-style-type: none">• El número de minutos especificado con el parámetro <code>summary</code> en el archivo de plantilla de exportación especificado (sólo <code>export</code>)• El intervalo de resumen predeterminado de una hora (<code>export</code> o <code>extract</code>) El resumen puede reducir en gran medida el tamaño de los datos extraídos o exportados resultantes, dependiendo del intervalo de resumen usado. Por ejemplo, los datos de resumen de cada hora representan aproximadamente una décima parte del tamaño de datos de detalles de 5 minutos.
<code>both</code> (sólo <code>export</code>)	Especifica si se extraen o exportan datos de detalle y de resumen.
<code>off</code>	Especifica que no se exportarán o extraerán datos de este tipo.



Si está usando Performance Manager, los datos de detalles deben incluirse en un archivo extraído antes de dibujar los gráficos de la aplicación con puntos cada 5 minutos.

Ejemplo

En este ejemplo, el comando `application` hace que los datos del archivo de registro detallados de la aplicación se exporten: El archivo `output export` contiene las métricas de la aplicación especificadas en el archivo de plantilla de exportación `myrept`.

```
logfile /var/opt/perf/datafiles/logglob
global off
application detail
report /var/opt/perf/myrept
export
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -a -r /var/opt/perf/myrept -xp
```

class

Use el comando `class` para especificar la clase de datos DSI que se exportará, o los datos `scope` que se extraerán o exportarán.

El valor predeterminado es `class detail`.

Sintaxis

```
class [nombre_clase] [detail]
      [summary]
      [both]
      [off]
```

Parámetros

`nombre_clase` Nombre de un grupo de métricas con una clasificación similar.

`detail` Para los archivos de registro DSI, especifica la cantidad de datos de detalle que se exportará de acuerdo con el tiempo establecido en el archivo de registro DSI. Para obtener más información, consulte [Introducción general de Integración de Orígenes de Datos \(DSI\) en la página 263](#).

Para los archivos de registro `scope`, especifica que los datos sin procesar, con los detalles de cinco minutos, deben extraerse o exportarse.

`summary` Consulte “Parámetros” en la descripción del comando `application` en la página 123. `Summary` y `both` sólo pueden exportarse.

Ejemplos

Para exportar datos de resumen en un archivo de registro DSI que contenga una clase denominada `acctg_info`, emita el siguiente comando:

```
class acctg_info summary
```

Una vez que el usuario ha especificado el archivo de registro, y que el programa `extract lo` ha abierto, se verifica que exista la clase `acctg_info` en el archivo de registro y que pueda exportarse.

Otras variaciones de este comando son:

```
CLASS ACCTG_INFO SUMMARY
class ACCTG_INFO summary
class acctg_info sum
```

Los comandos pueden escribirse en mayúsculas o minúsculas. Los nombres de clases siempre se ponen en mayúsculas y se comparan.

En el siguiente ejemplo se exportan los datos de resumen de la clase denominada `fin_info`.

```
extract>
class fin_info summary
export
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -l dsi.log -C fin_info summary -xp
```

configuration

Use el comando `configuration` para especificar si se exportará la información de configuración del sistema.

El valor predeterminado es `configuration off`.

Sintaxis

```
configuration      [on]
                    [detail]
                    [off]
```

Parámetros

`on` o `detail` Especifica que deben exportarse todos los registros de configuración.

`off` Especifica que no se exportará ningún dato de configuración.

Se exportará toda la información de configuración disponible en el archivo de registro. Se ignorará cualquier comando `begin`, `end`, `shift`, `start`, `stop` o `noweekends` usado con el comando `configuration`.



El comando `configuration` sólo afecta a la función `export`. La función `extract` no se ve afectada, ya que siempre extrae información de configuración del sistema.

Ejemplo

En este ejemplo, el comando `configuration` hace que la información de configuración del sistema se exporte. El archivo `output export` contiene las métricas de configuración especificadas en el archivo de plantilla de exportación `myrept`.

```
logfile /var/opt/perf/datafiles/logglob
configuration on
report /var/opt/perf/myrept
export
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -c -r /var/opt/perf/myrept -xp
```

cpu

Use el comando `cpu` para especificar el nivel de resumen de CPU.

El valor predeterminado es `cpu off`.

Sintaxis

```
cpu                [detail]
                    [summary]
                    [both] [off]
```

Parámetros

detail	Extrae o exporta registros de detalle de 5 minutos.
summary	Exporta registros de resumen.
both	Exporta registros de detalle y de resumen.
off	No extrae o exporta datos de CPU.

Ejemplo

En este ejemplo, el comando `cpu` hace que se exporten los datos de `detail` de CPU recopilados a partir del 26 de julio de 2001. Puesto que no se especifica ningún archivo de plantilla de exportación, se usa el predeterminado: `reptfile`. Todas las métricas de los discos se incluyen en el archivo de salida, como se especifica en `reptfile`.

```
logfile /var/opt/perf/datafiles/logglob
global off
cpu detail
start 7/26/01
export
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -u -b 7/26/01 -xp
```

disk

Use el comando `disk` para especificar el tipo de datos del dispositivo de disco que se extraerán o exportarán.

El valor predeterminado es `disk off`.

Sintaxis

```
disk      [on]
           [detail]
           [summary]
           [both]
           [off]
```

Parámetros

<code>on</code> o <code>detail</code>	Consulte “Parámetros” en la descripción del comando
<code>summary</code>	application al principio de este capítulo. <code>Summary</code> y <code>both</code>
<code>both</code> o <code>off</code>	sólo pueden exportarse.

Ejemplo

En este ejemplo, el comando `disk` hace que se exporten los datos de `disk detail` recopilados a partir del 5 de julio de 1999. Puesto que no se especifica ningún archivo de plantilla de exportación, se usa el predeterminado: `reptfile`. Todas las métricas de los discos se incluyen en el archivo de salida, como se especifica en `reptfile`.

```
logfile /var/opt/perf/datafiles/logglob
global off
disk detail
start 7/5/99
export
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -D -b 7/5/99 -xp
```

exit

Use el comando `exit` para finalizar el programa `extract`. El comando `exit` equivale al comando `extract` del programa `quit`.

Sintaxis

```
exit
e
```

export

Use el comando `export` para iniciar el proceso de copiar datos en un formato de archivo exportado.

Sintaxis

```
export      [day           [ddd] [aaddd] [-días]]
             [week       [ss] [aass] [-semanas]]
             [month     [mm] [aamm] [-meses]]
             [year      [aa] [aaaa] [-años]]
```

Parámetros

Use uno de los siguientes parámetros para exportar datos para un determinado intervalo.

<code>day</code>	Representa un solo día
<code>week</code>	Representa una sola semana, de lunes a domingo
<code>month</code>	Representa un solo mes, del primer al último día natural
<code>year</code>	Representa un solo año, del primer al último día natural

Si no se usa ningún parámetro con el comando `export`, el intervalo usado para los datos exportados se establece mediante los comandos `start` y `stop`.

Instrucciones sobre su uso

Hay cuatro formas de especificar un determinado intervalo (`day`, `week`, `month`, `year`).

- Intervalo actual: especifique sólo el parámetro. Por ejemplo, `month` significa el mes actual.
- Intervalo anterior: especifique el parámetro, un signo menos y el número de intervalos antes del actual deseado. Por ejemplo, `day-1` es ayer, `week-2` es dos semanas antes de la semana actual.
- Intervalo absoluto: especifique el parámetro y un número positivo. El número indica un intervalo absoluto deseado en el año actual. Por ejemplo, `day 2` es el 2 de enero del año actual.
- Intervalo absoluto más año: especifique el parámetro y un número positivo largo. El número debe estar formado por al menos dos dígitos del año y el número del intervalo absoluto de ese año. En este formato, el día absoluto tendría 5 dígitos (`99002` significa 2 de enero de 1999) y todos los demás parámetros tendrían cuatro dígitos (`month 9904` significa abril de 1999).

Si no ha especificado previamente un archivo de registro o un archivo de plantilla de exportación, el comando `logfile` usa el archivo de registro global predeterminado `logglob` y el comando `report` usa el archivo de plantilla de exportación predeterminado `reptfile`.

Se usa la configuración o los valores predeterminados de todos los demás parámetros. Para obtener información detallada acerca de sus acciones, consulte las descripciones de los comandos `application`, `configuration`, `global`, `process`, `disk`, `lvolume`, `netif`, `CPU`, `filesystem`, `transaction`, `output`, `shift`, `start` y `stop`.

El comando `export` crea hasta 16 archivos de salida predeterminados basados en los tipos de datos y el nivel de resumen especificado.

<code>xfrdGLOBAL.ext</code>	Archivo de datos de detalle globales
<code>xfrsGLOBAL.ext</code>	Archivo de datos de resumen global por horas
<code>xfrdAPPLICATION.ext</code>	Archivo de datos de detalle de aplicación
<code>xfrsAPPLICATION.ext</code>	Archivo de datos de resumen de aplicación por horas
<code>xfrdPROCESS.ext</code>	Archivo de datos de detalle de proceso
<code>xfrdDISK.ext</code>	Archivo de datos de detalle de dispositivo de disco
<code>xfrsDISK.ext</code>	Archivo de datos de resumen de dispositivo de disco
<code>xfrdVOLUME.ext</code>	Archivo de datos de detalle del volumen lógico
<code>xfrsVOLUME.ext</code>	Archivo de datos de resumen del volumen lógico
<code>xfrdNETIF.ext</code>	Archivo de datos de detalle de NETIF
<code>xfrsNETIF.ext</code>	Archivo de datos de resumen de NETIF

xfrdCPU.ext	Tipo de datos de detalle de CPU
xfrsCPU.ext	Tipo de datos de resumen de CPU
xfrdFILESYSTEM.ext	Tipo de datos de detalle del sistema de archivos
xfrsFILESYSTEM.ext	Tipo de datos de resumen del sistema de archivos
xfrdTRANSACTION.ext	Archivo de datos de detalle de transacción
xfrsTRANSACTION.ext	Archivo de datos de resumen de transacción
xfrdCONFIGURATION.ext	Archivo de datos de detalle de configuración

donde ext = asc, dat, bin o wk1

Los nombres de archivo predeterminados se crean a partir del nombre del tipo de datos. El prefijo es `xfrd` o `xfrs` dependiendo de si los datos son datos de detalle o de resumen. La extensión corresponde al formato de datos `asc` (ASCII), `bin` (binario), `dat` (archivo de datos) o `wk1` (hoja de cálculo).

Por ejemplo, nombre_clase = ACCTG_INFO tendría los siguientes nombres de archivo de export:

xfrdACCTG_INFO.wk1 Datos de la hoja de cálculo detallados para ACCT_INFO

xfrsACCTG_INFO.asc Datos ASCII resumidos para ACCT_INFO

Para obtener más información sobre la exportación de datos, consulte [Introducción general de la función de exportación](#) en la página 102 en el Capítulo 5.

Ejemplo

En este ejemplo, el comando `export` hace que se exporten los datos del archivo de registro recopilados ayer entre las 8:00 y las 5 pm. Puesto que no se especifica ningún archivo de plantilla de exportación, se usa el predeterminado: `reptfile`. Todas las métricas globales se incluyen en el archivo de salida, como se especifica en `reptfile`.

```
logfile /var/opt/perf/datafiles/logglob
start today-1 8:00 am
stop today-1 5:00 pm
global both
export
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -g -b today-1 8:00 am -e today-1 5:00 pm -xp
```

extract

Use el comando `extract` para iniciar el proceso de copiar datos del archivo de registro sin procesar en un formato de archivo extraído. Los archivos extraídos pueden usarse para archivar o para que programas como Performance Manager realicen análisis. Puede extraer datos de archivos de registro sin procesar y de archivos extraídos.

El comando `extract` no puede usarse para procesar datos de los archivos de registro DSI.

Sintaxis

```
extract    [day          [ddd] [aaddd] [-días]]
           [week       [ss] [aass] [-semanas]]
           [month      [mm] [aamm] [-meses]]
           [year       [aa] [aaaa] [-años]]
```

Parámetros

Use uno de los siguientes parámetros para extraer datos para un determinado intervalo:

<code>day</code>	Representa un solo día
<code>week</code>	Representa una sola semana, de lunes a domingo
<code>month</code>	Representa un solo mes, del primer al último día natural
<code>year</code>	Representa un solo año, del primer al último día natural

Si no se usa ningún parámetro con el comando `extract`, el intervalo usado para la extracción de datos se establece mediante los comandos `start` y `stop`.

Instrucciones sobre su uso

Hay cuatro formas de especificar un determinado intervalo (`day`, `week`, `month`, `year`).

- Intervalo actual: especifique sólo el parámetro. Por ejemplo, `month` significa el mes actual.
- Intervalo anterior: especifique el parámetro, un signo menos y el número de intervalos antes del actual deseado. Por ejemplo, `day-1` es ayer, `week-2` es dos semanas antes de la semana actual.
- Intervalo absoluto: especifique el parámetro y un número positivo. El número indica un intervalo absoluto deseado en el año actual. Por ejemplo, `day 2` es el 2 de enero del año actual.
- Intervalo absoluto más año: especifique el parámetro y un número positivo largo. El número debe estar formado por al menos dos dígitos del año y el número del intervalo absoluto de ese año. En este formato, el día absoluto tendría 5 dígitos (99002 significa 2 de enero de 1999) y todos los demás parámetros tendrían cuatro dígitos (`month 99904` significa abril de 1999).

El comando `extract` inicia la extracción de datos. Si no se ha especificado previamente, los comandos `logfile` y `output` asumen los siguientes valores predeterminados cuando se ejecuta el comando `extract`:

```
log file = /var/opt/perf/datafiles/logglob
output file = rxlog,new
```

Se usa la configuración o los valores predeterminados de todos los demás parámetros. Para obtener información detallada acerca de sus acciones, consulte las descripciones de los comandos `application`, `global`, `process`, `disk`, `lvolume`, `netif`, `CPU`, `filesystem`, `transaction`, `shift`, `start` y `stop`.

El tamaño de un archivo de registro extraído no puede exceder de 3,5 gigabytes.

Ejemplo

En el primer ejemplo, se extraen los datos recopilados desde el 1 de marzo de 2000 hasta el 30 de junio de 2000 durante las horas entre 8:00 am y 5:00 pm los días laborables. Sólo se extraen los datos de detalle de la aplicación y globales.

```
logfile /var/opt/perf/datafiles/logglob
start 03/01/00
stop 06/30/00
shift 8:00 am - 5:00 pm nowweekends
global detail
application detail
extract
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -ga -b 03/01/00 -e 6/30/00 -s 8:00 am - 5:00 nowweekends -xt
```

En el segundo ejemplo se crea un archivo de registro extraído denominado rxjan00. Se depura cualquier archivo existente que tenga este nombre. Se extraen todos los datos del archivo de registro sin procesar recopilados entre el 1 de enero de 2000 y el 31 de enero de 2000:

```
logfile /var/opt/perf/datafiles/logglob
output rxjan00,purge
start 01/01/00
stop 01/31/00
global detail
application detail
transaction detail
process detail
disk detail
lvolume detail
netif detail
filesystem detail
cpu detail
extract
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -f rxjan00,purge -gatpdznyu -b 01/01/00 -e 01/31/00 -xt
```

filesystem

Use este comando para especificar el nivel de resumen de los datos del sistema de archivos en `extract 0 export`.

El valor predeterminado es `filesystem off`.

Sintaxis

```
filesystem    [detail]
               [summary]
               [both]
               [off]
```

Parámetros

detail	Extrae o exporta registros de detalle de 5 minutos.
summary	Exporta registros de resumen.
both	Exporta registros de detalle y de resumen.
off	No extrae o exporta datos del sistema de archivos.

Ejemplo

En este ejemplo, el comando `filesystem` hace que se exporten los datos de detalle del sistema de archivos recopilados a partir del 26 de julio de 2001. Puesto que no se especifica ningún archivo de plantilla de exportación, se usa el predeterminado: `reptfile`. Todas las métricas del sistema de archivos se incluyen en el archivo de salida, como se especifica en `reptfile`.

```
logfile /var/opt/perf/datafiles/logglob
global off
filesystem detail
start 7/26/01
export
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -y -b 7/26/01 -xp
```

global

Use el comando `global` para especificar la cantidad de datos globales que se extraerán o exportarán.

El valor predeterminado es `global detail`. (En el modo de la línea de comandos, el valor predeterminado es `global off`).

Sintaxis

```
global [on]
         [detail]
         [summary]
         [both]
         [off]
```

Parámetros

detail u on	Consulte “Parámetros” en la descripción del comando
summary	application al principio de este capítulo. Summary y both
both	sólo pueden exportarse.
off	

Instrucciones sobre su uso

Los datos de detalle deben extraerse si desea dibujar gráficos globales de Performance Manager con puntos cada cinco minutos.

Performance Manager realiza un gráfico de los datos resumidos más rápidamente, ya que son necesarios menos registros de datos para generar un gráfico. Si sólo se extraen los resúmenes globales, los gráficos globales de Performance Manager no pueden dibujarse con puntos de datos cada cinco minutos.

La opción `both` mantiene la velocidad de acceso obtenida con los registros de resumen por horas al tiempo que permite dibujar gráficos globales de Performance Manager con puntos cada cinco minutos.

No se recomienda el parámetro `off` si está usando Performance Manager, ya que debe tener datos globales para comprender adecuadamente el comportamiento global del sistema. Los gráficos globales de Performance Manager no pueden dibujarse a menos que el archivo extraído contenga al menos un tipo de datos globales.

Ejemplo

El comando `global` se usa aquí para especificar que *no* se exporten datos globales (el valor predeterminado es `global detail`). Sólo se exportan los datos de transacciones detalladas. El archivo `output export` contiene las métricas de la transacción especificadas en el archivo de plantilla de exportación `myrept`.

```
extract>
logfile /var/opt/perf/datafiles/logglob
global off
transaction detail
report /var/opt/perf/myrept
export
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -l -t -r /var/opt/perf/myrept -xp
```

guide

El comando `guide` permite introducir el modo de comandos guiados. La interfaz de comandos guiados conduce al usuario por los diversos comandos `extract` y solicita al usuario que realice algunas de las tareas más comunes que estén disponibles.

Sintaxis

guide

Instrucciones sobre su uso

- Para introducir el modo de comandos guiado en el modo interactivo de `extract`, escriba **guide**.
- Para aceptar el valor predeterminado correspondiente a un parámetro, el usuario deberá pulsar **Entrar**.
- Para finalizar el modo de comandos guiados y regresar al modo interactivo, el usuario deberá escribir **q** tras el símbolo del sistema `guide>`.

Este comando no ofrece todas las combinaciones posibles de configuración de parámetros. Selecciona la configuración que debería producir resultados útiles para la mayoría de los usuarios. Puede obtener un control completo sobre las funciones de `extract` mediante el modo de comandos interactivo de `extract`.



Si está exportando datos de archivos del registro DSI, recomendamos usar el modo de comandos guiados para crear un archivo de plantilla de exportación personalizado y exportar los datos.

help

Use el comando `help` para acceder a la ayuda en línea.

Sintaxis

```
help [palabraclave]
```

Instrucciones sobre su uso

Pueden introducirse parámetros para obtener información sobre las tareas y los comandos `extract`, o en `help` mismo. Se puede navegar por los diferentes temas introduciendo una palabra clave. Si se encuentra disponible más de una página de información, la pantalla se detiene y se espera a que el usuario pulse **Entrar** para continuar. Escriba **q** o **quit** para salir del sistema de ayuda y regresar al programa `extract`.

También se puede pedir ayuda sobre un tema determinado. Por ejemplo:

```
help tasks
```

o bien

```
help resize parms
```

Cuando se use este modo del comando `help`, el usuario recibirá el texto de ayuda para el tema especificado y permanecerá en el contexto de entrada de comando `extract`. Como no se introduce el subsistema de ayuda de forma interactiva, no se tiene que escribir **quit** antes de introducir el siguiente comando `extract`.

list

El comando `list` permite especificar el archivo de lista de todos los informes del programa `extract`.

Sintaxis

```
list  [archivo  
      ]  
      [*]
```

Instrucciones sobre su uso

Puede usar `list` en cualquier momento cuando utiliza `extract` para especificar el dispositivo de la lista. Usa un nombre de archivo o el nombre del dispositivo de la lista para generar la salida de la configuración especificada por el usuario. Si ya existe el archivo de la lista, se le anexa la salida.

Los datos enviados al dispositivo de la lista también se muestran en la pantalla.

Mientras `extract` se está ejecutando, escriba:

```
list outfilename
```

Para devolver el dispositivo de la lista al terminal del usuario, escriba:

```
list stdout
```

o bien

```
list *
```

Para determinar el dispositivo de la lista actual, escriba el comando `list` sin parámetros, como se muestra a continuación:

```
list
```

Si el archivo de la lista no es `stdout`, la mayoría de los comandos serán reproducidos en el archivo de la lista conforme sean introducidos.

Ejemplo

En el siguiente ejemplo, el dispositivo de la lista se establece en `mylist`. El resultado de los siguientes comandos se imprime en `mylist` y se muestra en la pantalla.

```
extract>
logfile /var/opt/perf/datafiles/logglob
list mylist
global detail
shift 8:00 AM - 5:00 PM
extract
```

logfile

El comando `logfile` permite abrir un archivo de registro. Debe abrir un archivo de registro para todas las funciones del programa `extract`. Puede hacerlo explícitamente emitiendo el comando `logfile` o implícitamente emitiendo el comando `extract` o el comando `export`. Si no especifica un nombre de archivo de registro, el programa `extract` le solicita el nombre de un archivo de registro y muestra el archivo de registro global predeterminado `/var/opt/perf/datafiles/logglob`. Se puede aceptar el nombre predeterminado o especificar otro archivo de registro.

Sintaxis

```
logfile [archivo_registro]
```

Instrucciones sobre su uso

Para abrir un archivo de registro, puede especificar el nombre de un archivo de registro extraído o sin procesar. No se puede especificar el nombre de un archivo creado por el comando `export`. Si se especifica el nombre del archivo de registro extraído, toda la información será obtenida de este mismo archivo. Si especifica un nombre de archivo de registro sin procesar, debe especificar el nombre del archivo de registro global antes de acceder al archivo de registro sin procesar. Es el único nombre de archivo de registro sin procesar que debe especificar.

Si el archivo de registro no está en su directorio del trabajo actual, debe proporcionar su ruta.

El archivo de registro global y otros archivos de registro sin procesar deben estar en el mismo directorio. Tienen los siguientes nombres:

logglob	archivo de registro global
logappl	archivo de registro de aplicación
logproc	archivo de registro de proceso
logdev	archivo de registro de dispositivo
logtran	archivo de registro de transacción
logindx	archivo de registro de índice

El contenido general del archivo de registro se muestra al abrir el archivo de registro.



No cambie el nombre de los archivos de registro sin procesar. Al acceder a estos archivos, el programa considera que los nombres de archivo de registro estándares están en vigor. Si debe cambiar el nombre de los archivos de registro para colocar archivos de registro de varios sistemas en el mismo sistema para su análisis, antes debe extraer los datos y, a continuación, cambiar el nombre de los archivos de registro extraídos.

Ejemplo

A continuación se muestra un listado típico del archivo de registro global predeterminado.

```
Global      file: /var/opt/perf/datafiles/logglob, version D
Application file: /var/opt/perf/datafiles/logappl
Process     file: /var/opt/perf/datafiles/logproc
Device      file: /var/opt/perf/datafiles/logdev
Transaction file: /var/opt/perf/datafiles/logdev
Index       file: /var/opt/perf/datafiles/logindx
System ID:  homer
System Type 9000/715/ S/N 2223334442 O/S HP-UX B.10.20 A
Data collector: SCOPE/UX C.02.30
File Created: 10/08/99
Data Covers: 44 days to 11/20/99
Shift is:    All Day
Data records available are:
Global Application Process Disk Volume Transaction
Maximum file sizes:
Global=10.0 Application=10.0 Process=20.0 Device=10.0 Transaction=10.0 MB
The first GLOBAL      record is on 10/08/99 at 08:17 AM
The first NETIF       record is on 10/08/99 at 08:17 AM
The first APPLICATION record is on 11/17/99 at 12:15 PM
The first PROCESS     record is on 10/08/99 at 08:17 AM
The first DEVICE      record is on 10/31/99 at 10:45 AM
The Transaction data file is empty
The default starting date & time = 10/08/99 11:50 AM (LAST -30)
The default stopping date & time = 11/20/99 11:59 PM (LAST -0)
```


lvolume

Use el comando `lvolume` para especificar el tipo de datos del volumen lógico que se extraerán o exportarán. (Sólo los sistemas HP-UX usan este comando).

El valor predeterminado es `lvolume off`.

Sintaxis

```
lvolume    [on]
           [detail]
           [summary]
           [both]
           [off]
```

Parámetros

`on` o `detail` Consulte “Parámetros” en la descripción del comando
`summary` [application](#) al principio de este capítulo. `Summary` y `both`
`both` sólo pueden exportarse.
`off`

Ejemplo

En este ejemplo, se crea un archivo de registro extraído denominado `rx899` y se depura cualquier archivo existente con ese nombre. Se extraen todos los datos de volumen lógicos recopilados entre el 1 de agosto y el 31 de agosto.

```
logfile /var/opt/perf/datafiles/logglob
output rx899,purge
start 08/01/99
stop 08/31/99
global detail
lvolume detail
month 9908
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -f rx899,purge -gz -xm 9908
```

menu

Use el comando `menu` para imprimir una lista de los comandos `extract` disponibles.

Sintaxis

```
menu
```

Ejemplo

Command	Parameters	Function
HELP	[topic]	Get information on commands and options
GUIDE		Enter guided commands mode for novice users
LOGFILE	[logname]	Specify a log file to be processed
LIST	[filename *]	Specify the listing file
OUTPUT	[filename] [,NEW/PURGE/APPEND]	Specify a destination file
REPORT	[filename][,SHOW]	Specify an Export Format file for EXPORT"
GLOBAL	[DETAIL/SUMMARY/BOTH/OFF]	Extract GLOBAL records
APPLICATION	[DETAIL/SUMMARY/BOTH/OFF]	Extract APPLICATION records
PROCESS	[DETAIL/OFF/KILLED][APP=]	Extract PROCESS records
DISK	[DETAIL/SUMMARY/BOTH/OFF]	Extract DISK DEVICE records
LVOLUME	[DETAIL/SUMMARY/BOTH/OFF]	Extract Logical VOLUME records
NETIF	[DETAIL/SUMMARY/BOTH/OFF]	Extract Logical NETIF records
CPU	[DETAIL/SUMMARY/BOTH/OFF]	Extract CPU records
FILESYSTEM	[DETAIL/SUMMARY/BOTH/OFF]	Extract FILESYSTEM records
CONFIG	[DETAIL/OFF]	Export CONFIGURATION records
CLASS	classname[DETAIL/SUMMARY/BOTH/OFF]	Export classname records
TRANSACTION	[DETAIL/SUMMARY/BOTH/OFF]	Extract TRANSACTION records
START	[startdate time]	Specify a starting date and time for SCAN
STOP	[stopdate time]	Specify an ending date and time for SCAN
SHIFT	[starttime - stoptime] [NOWEEKENDS]	Specify daily shift times
SHOW	[ALL]	Show the current program settings
EXPORT	[d/w/m/y][-offset]	Copy log file records to HOST format files
EXTRACT	[d/w/m/y][-offset]	Copy selected records to output (or append) file
WEEKLY	[ww/yyww]	Extract one calendar week's data with auto file names
MONTHLY	[mm/yymm]	Extract one calendar month's data with auto file names
YEARLY	[yy/yyyy]	Extract one calendar year's data with auto file names
WEEKDAYS	[1...7]	Set days to exclude from export 1=Sunday ! or SH
MENU or ?		List the command menu (this listing)
EXIT or Q		Terminate the program

monthly

Use el comando `monthly` para especificar la extracción de datos basada en un mes natural. Durante la ejecución, este comando establece las fechas de inicio y parada en las flechas adecuadas, según el mes y el año de los datos extraídos.

El nombre del archivo de salida está formado por las letras `rxmo` seguidas por los cuatro dígitos del año y un número de dos dígitos del mes de la extracción. Por ejemplo, los datos extraídos en marzo de 1999 se colocarían en un archivo denominado `rxmo199903`.

Sintaxis

```
monthly    [aamm]  
            [mm]
```

Parámetros

<code>monthly</code>	Extrae datos del mes actual (predeterminado).
<code>monthly mm</code>	Extrae datos para un mes específico desde los datos del año actual (donde <i>mm</i> es un número entre 01 y 12).
<code>monthly aamm</code>	Extrae datos de un mes y año específicos (donde <i>aamm</i> es un solo número formado por los últimos dos dígitos del año y los dos dígitos del mes). Por ejemplo, para extraer los datos de febrero de 1999, especifique <code>monthly 9902</code> .

Si no especifica el archivo de registro antes de ejecutar el comando `monthly`, se usa el archivo `logglob` predeterminado.

Instrucciones sobre su uso

Use el comando `monthly` cuando está extrayendo datos para archivarlos mensualmente.

El tipo de datos extraídos y resumidos sigue las reglas normales del comando `extract` y puede establecerse antes de ejecutar el comando `monthly`. Se usa esta configuración a menos que ya exista un archivo de salida mensual. En este caso, los datos se anexan a él según el tipo de datos especificado originalmente.

El comando `monthly` tiene una función que abre el archivo extraído del mes anterior y comprueba si está lleno; si contiene datos extraídos hasta el último día del mes. Si no es así, el comando `monthly` anexa datos a este archivo para completar la extracción del mes anterior.

Por ejemplo, se ejecuta un comando `monthly` el 7 de mayo de 1999. Esto crea un archivo de registro denominado `rxmo199905` que contiene datos desde el 1 de mayo hasta la fecha actual (7 de mayo).

El 4 de junio de 1999, se ejecuta otro comando `monthly`. Antes de que se cree el archivo `rxmo199906` para el mes actual, se abre y comprueba el archivo `rxmo199905` del mes anterior. Cuando se encuentra que está incompleto, se anexan a él los datos para completar la extracción hasta el 31 de mayo de 1999. A continuación, se crea el archivo `rxmo199906` para que almacene los datos desde el 1 de junio de 1999 hasta la fecha actual (4 de junio).

Mientras se ejecute el comando `monthly` al menos una vez al mes, esta característica completará cada archivo del mes antes de crear el archivo del siguiente mes. Al ver dos archivos mensuales adyacentes; por ejemplo, `rxmo199905` (mayo) y `rxmo199906` (junio), puede suponer que el primer archivo está completo para ese mes y que puede archivarse y depurarse.



Los comandos `monthly` y `extract month` son similares en el sentido de que los dos extraen los datos de un mes natural. El comando `monthly` ignora la configuración del comando `output` y usa en su lugar los nombres de los archivos de salida preferidos. También intenta analizar los datos que faltan al archivo de registro extraído del mes anterior si aún está presente en el sistema. Por otra parte, el comando `extract month`, usa la configuración del comando `output`. No puede anexas datos al archivo extraído del mes anterior, ya que no conoce su nombre.

Ejemplo

En este ejemplo, se extraen los datos de detalle de aplicaciones registrados durante mayo de 1999.

```
logfile /var/opt/perf/datafiles/logglob
global off
application detail
monthly 9905
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -a -xm 9905
```

netif

Use el comando `netif` para especificar el tipo de datos de la interfaz de red lógica (LAN) para `extract` o `export`. Los datos de Netif se registran en el archivo `logdev`.

El valor predeterminado es `netif off`.

Sintaxis

```
netif    [on]
         [detail]
         [summary]
         [both]
         [off]
```

Parámetros

<code>on</code> o <code>detail</code>	Consulte “Parámetros” en la descripción del comando
<code>summary</code>	application al principio de este capítulo. <code>Summary</code> y <code>both</code>
<code>both</code>	sólo pueden exportarse.
<code>off</code>	

Ejemplo

En este ejemplo, se extraen los datos de detalle de `netif` recopilados desde el 1 de marzo de 2000 hasta el 30 de junio de 2000 durante las horas entre 8:00 am y 5:00 pm los días laborables.

```
logfile /var/opt/perf/datafiles/logglob
start 03/01/00
stop 06/30/00
shift 8:00 AM - 5:00 PM noweekends
netif detail
extract
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -n -b 03/01/00 -e 6/30/00 -s 8:00 am - 5:00 noweekends -xt
```

output

Use el comando `output` para especificar el nombre de un archivo de salida para las funciones `extract` o `export`.

El segundo parámetro opcional especifica la acción que se realizará si existe un archivo de salida con el mismo nombre.

Sintaxis

```
output    [nombre_archi    [,new]  
           vo]              [,purge]  
                           [,append]
```

Parámetros

- ,**new** Especifica que el archivo de salida debe ser un archivo nuevo. Ésta es la acción predeterminada en el modo por lotes. Si existe un archivo con el mismo nombre, el trabajo por lotes termina.
- ,**purge** Especifica que se depure un archivo existente para que haya espacio para el nuevo archivo de salida.
- ,**append** Especifica que se anexen los datos a un archivo extraído existente. Si no hay ningún archivo con el nombre de archivo de salida especificado, se crea un archivo.

Instrucciones sobre su uso

Si no especifica ninguna acción en el modo por lotes, se usa la opción predeterminada `,new`. En el modo interactivo, se le solicita que introduzca una acción si se encuentra un archivo duplicado.

Si no especifica un archivo de salida, se crean archivos de salida predeterminados. Los nombres de los archivos de salida predeterminados son:

Para `extract`: `rxlog`

Para `export`:

```
xfrdGLOBAL.ext  
xfrsGLOBAL.ext  
xfrdAPPLICATION.ext  
xfrsAPPLICATION.ext  
xfrdPROCESS.ext  
xfrdDISK.ext  
xfrsDISK.ext  
xfrdLVOLUME.ext  
xfrsLVOLUME.ext  
xfrdNETIF.ext  
xfrsNETIF.ext  
xfrdCPU.ext  
xfrsCPU.ext  
xfrdFILESYSTEM.ext
```

```
xfrsFILESYSTEM.ext
xfrdTRANSACTION.ext
xfrsTRANSACTION.ext
xfrdCONFIGURATION.ext
```

donde `ext` = `asc` (ASCII), `dat` (archivo de datos), `bin` (binario) o `wk1` (hoja de cálculo).

Se puede usar un nombre de archivo especial, `stdout` (o `*`), con la operación de exportación para dirigir la salida al archivo `stdout` (normalmente el terminal o la estación de trabajo, aunque puede redirigirse mediante los comandos shell).

output stdout

o bien

output *

Para devolver la salida a su configuración predeterminada, escriba:

output default

o bien

output -



Puede sobrescribir los nombres de los archivos de salida predeterminados mediante el parámetro `output` del archivo de plantillas de exportación.

No debería colocar la salida de los archivos de funcionamiento de extracción en `stdout`, ya que son incompatibles con los dispositivos ASCII. Tampoco debería colocar los formatos de salida binaria WK1 de la operación de exportación en el archivo `stdout` por la misma razón.

Es necesario tener cuidado para evitar que se añadan los datos *extraídos* a un archivo de datos *exportado* existente y para evitar que señalen los datos exportados a un archivo extraído existente. Los intentos de anexar un tipo de datos incorrecto pueden producir un error.

Ejemplo

En este ejemplo, no se especifica ningún archivo de salida, por lo que se usa el archivo de salida predeterminado, `rxlog`, para los datos de resumen de la aplicación que se están extrayendo. La opción `,purge` especifica que debe depurarse cualquier archivo de salida existente.

```
extract>
logfile /var/opt/perf/datafiles/logglob
output rxlog,purge
global off
application detail
extract month 9905
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

extract -f rxlog,purge -a -xm 9905

process

Use el comando `process` para especificar si se extraerán o exportarán los datos del proceso.

El valor predeterminado es `process off`.

Sintaxis

```
process [on] [detail] [application=#[-#] ,...]  
[off]  
[killed]
```

Parámetros

on	Especifica que los datos de proceso <i>deben</i> extraerse o exportarse.
detail	Especificar process detail es igual a especificar process on.
off	Especifica que los datos de proceso <i>no deben</i> extraerse o exportarse.
killed	Sólo especifica procesos que tienen una razón de interés que incluya killed. (Procesos terminados en el intervalo de medición).
application	Sólo especifica procesos que pertenecen a aplicaciones seleccionadas. Una aplicación se puede introducir como un solo número o como una serie de números de aplicación (7-9 significa las aplicaciones 7, 8 y 9). El número de aplicación está determinado por el orden de la definición de la aplicación en el archivo de parámetros cuando se recopilaron los datos. Si se especifican varias aplicaciones, separe cada una con una coma.



Los datos de proceso pueden aumentar el tamaño de un archivo de registro extraído en gran medida. Si piensa copiar el archivo de registro en una estación de trabajo para su análisis, puede desear limitar la cantidad de datos de procesos extraídos.

Ejemplo

En este ejemplo, el comando process especifica procesos terminados durante un intervalo, y pertenece las aplicaciones 1, 4, 6, 7, 8 o 10. Use el comando scan del programa utility para encontrar los números de aplicación para determinadas aplicaciones.

```
process killed applications=1,4,6-8,10
```

quit

Use el comando quit para finalizar el programa extract. El comando quit equivale al comando exit del programa extract.

Sintaxis

```
quit  
q
```

report

Use el comando `report` para especificar el archivo de la plantilla de exportación utilizado por la función `export`. Si no se especifica ningún archivo de plantilla de exportación, se usa el predeterminado: `reptfile`. El archivo de plantilla de exportación se usa para especificar diversos atributos de formato de salida usados en la función `export`. También especifica las métricas que se exportarán.

Si esta en modo interactivo y no especifica ningún archivo de plantilla de exportación, todas las métricas de los tipos de datos solicitados se exportarán en formato ASCII.

Sintaxis

```
report [archivo_plantilla_exportación] [, show]
```

Parámetros

`,show` Especifica que las posiciones de campo y las columnas de inicio deben enumerarse para todas las métricas especificadas en el archivo de plantilla de exportación. Este listado puede usarse cuando otros programas procesan los archivos de exportación.

Instrucciones sobre su uso

Al emitir este comando, un mensaje le pregunta si desea validar las métricas de la plantilla de exportación con el archivo de registro antes especificado. La validación garantiza que las métricas especificadas en el archivo de plantilla de exportación existan en el archivo de registro. Esto permite buscar posibles errores en el archivo de plantilla de exportación. Si no se realiza ninguna validación, esta acción se pospone hasta que realice una exportación.



El parámetro `,show` del comando `report` es distinto del comando `show` comentado más adelante.

sh

`sh` permite introducir un comando shell sin salir de `extract` escribiendo `sh` o un signo de exclamación (!) seguido del comando shell de UNIX.

Sintaxis

```
sh o ! [shell command]
```

Parámetros

`sh ls` Ejecuta el comando `ls` y vuelve a `extract`. El comando shell es cualquier comando del sistema.

`!ls` Idem.

`!ksh` Inicia un shell Korn. No vuelve inmediatamente a `extract`. Escriba `exit` o `CTRL-d Entrar` para volver al programa `extract`.

Instrucciones sobre su uso

Después de la ejecución del comando concreto, se regresa a `extract`. Si el usuario desea emitir múltiples comandos shell sin regresar a `extract` después de cada uno de ellos, podrá iniciar un nuevo shell.

Si emite el comando `sh` sin el nombre del comando shell, se le solicita que lo proporcione. Por ejemplo:

```
sh
enter SYSTEM command: ls
```

shift

Use el comando `shift` para limitar la extracción de datos registrados a determinadas horas del día que se correspondan a turnos de trabajo y excluir fines de semana (sábado y domingo).

El valor predeterminado es `shift all day` para extraer datos para todo el día, todos los días, incluidos fines de semana.

Sintaxis

```
shift      [hora_inicio-hora_fin]
            [all day]
            [noweekends]
```

Parámetros

Los parámetros `hora_inicio` y `hora_fin` se introducen en el mismo formato que la hora en el comando `start`. Se permiten turnos que se amplían hasta la medianoche. Si se programa `hora_inicio después de hora_fin`, el turno empezará en la hora de inicio y continuará hasta después de la medianoche, terminando en `hora_fin` del día siguiente.

<code>all day</code>	Especifica el turno predeterminado de 12:00 am - 12:00 am (o 00:00 -00:00 en un reloj de 24 horas).
<code>noweekends</code>	Especifica la exclusión de datos registrados los sábados y domingos. Si se introduce <code>noweekends</code> junto con un turno que llega hasta la medianoche, el fin de semana estará compuesto por los turnos que <i>empiezan</i> el sábado o el domingo.

Ejemplo

En este ejemplo, se extraen los datos de detalle del disco recopilados entre las 10:00 am y las 4:00 pm todos los días a partir del 15 de junio de 1999.

```
extract>
logfile /var/opt/perf/datafiles/logglob
global off
disk detail
shift 10:00 am - 4:00 PM
start 6/15/99
extract
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -d -b 6/15/99 -s 10:00 AM-4:00 PM -xt
```

show

El comando `show` enuncia los nombres de los archivos abiertos y el estado de los parámetros de `extract` que pueden establecerse.

Sintaxis

```
show [all]
```



El parámetro `show` de este comando es distinto del parámetro `,show` del comando `report` comentado antes.

Ejemplos

Use `show` solo para generar una lista con un aspecto similar al siguiente:

```
Logfile: /var/opt/perf/datafiles/logglob
Output:  Default
Report:  Default
List:    "stdout"
The default starting date & time = 10/08/99 12:00 AM (LAST -30)
The default stopping date & time = 11/20/99 11:59 PM (LAST -0)
The default shift = 12:00 AM - 12:00 PM
GLOBAL          DETAIL          records will be processed
APPLICATION. . . . . NO records will be processed
PROCESS . . . . . NO records will be processed
DISK DEVICE. . . . . NO records will be processed
LVOLUME. . . . . NO records will be processed
TRANSACTION. . . . . NO records will be processed
NETIF . . . . . .NO records will be processed
CPU . . . . . .NO records will be processed
FILESYSTEM. . . . . .NO records will be processed
Configuration . . . . . .NO records will be processed
Use show all to produce a more detailed list that may look like this:
Logfile: /var/opt/perf/datafiles/logglob
Global   file: /var/opt/perf/datafiles/logglob,version D
Application file: /var/opt/perf/datafiles/logappl
Process  file: /var/opt/perf/datafiles/logproc
Device   file: /var/opt/perf/datafiles/logdev
Transaction file: /var/opt/perf/datafiles/logdev
Index    file: /var/opt/perf/datafiles/logindx
System ID: homer
System Type 9000/715/ S/N 2223334442 O/S HP-UX B.10.20 A
Data collector: SCOPE/UX C.02.30
File Created: 10/08/99
Data Covers: 44 days to 11/20/99
Shift is: All Day
Data records available are:
Global Application Process Disk Volume Transaction
Maximum file sizes:
Global=10.0 Application=10.0 Process=20.0 Device=10.0
Transaction=10.0 MB
Output: Default
Report: Default
List: "stdout"
```

```

The default starting date & time = 10/08/99 11:50 AM (LAST -30)
The default stopping date & time = 11/20/99 11:59 PM(LAST - 0)
The default shift = 12:00 AM - 12:00 PM
GLOBAL.....DETAIL.....records will be processed
APPLICATION.....NO records will be processed
PROCESS.....NO records will be processed
DISK DEVICE.....NO records will be processed
LVOLUME.....NO records will be processed
TRANSACTION.....NO records will be processed
NETIF.....NO records will be exported
CPU.....NO records will be processed
FILESYSTEM.....NO records will be processed
Configuration .....NO records will be exported
Export Report Specifications:
  Interval = 3600, Separator = " "
  Missing data will not be displayed
  Headings will be displayed
  Date/time will be formatted
  Days to exclude: None

```

start

Use el comando `start` para establecer una fecha y hora de inicio para las funciones `extract` y `export`. La fecha de inicio predeterminada es 30 días completos antes de la última fecha del archivo de registro, o bien si hay menos de 30 días, la fecha del registro más antigua del archivo de registro.

Sintaxis

```

start      [date [hora]]
             [today [-día][hora]]
             [last [-días][hora]]
             [first [+días][hora]]

```

Parámetros

date	El formato de <code>date</code> dependerá del idioma original configurado en el sistema. Si el usuario no usa ningún idioma original o ha establecido C como idioma predeterminado, el formato de fecha será <i>mm/dd/aa</i> (mes/día/año), por ejemplo, 09/30/99 en el caso del 30 de septiembre de 1999, para la función <code>extract</code> o <code>export</code> .
time	El formato de <code>time</code> dependerá también del idioma original en uso. En el caso de C, el formato es <i>hh:mm am</i> o <i>hh:mm pm</i> (hora:minuto en formato de 12 horas con el sufijo <i>am/pm</i>). Por ejemplo, 07:00 am son las 7 en punto de la mañana. Se acepta para todos los idiomas la especificación de hora en el formato de 24 horas. Por ejemplo, 23:30 serían las 11.30 pm. Si se introduce la fecha u hora en un formato inaceptable, se mostrará un ejemplo con el formato correcto. Si no se proporciona ninguna hora de inicio, se propondrá la medianoche (12:00 am). La hora de inicio de medianoche de un día dado será el <i>comienzo</i> de ese día (00:00 en un formato de 24 horas).
today	Especifica el día actual. La cualificación del parámetro, como <code>today-días</code> , especifica el número de días con <i>anterioridad</i> a la fecha de hoy. Por ejemplo, <code>today-1</code> indica la fecha de ayer y <code>today-2</code> la fecha de antesdeayer.
last	Puede usarse para representar la última fecha contenida en el archivo de registro. El parámetro <code>last-días</code> especifica el número de días con <i>anterioridad</i> a la última fecha contenida en el archivo de registro.
first	Puede usarse para representar la primera fecha contenida en el archivo de registro. El parámetro <code>first+días</code> especifica el número de días <i>tras</i> la última fecha contenida en el archivo de registro.

Instrucciones sobre su uso

Los siguientes comandos sobrescriben la fecha de inicio establecida por el comando `start`.

- `weekly`
- `monthly`
- `yearly`
- `extract` (si se usan los parámetros `day`, `week`, `month` o `year`)
- `export` (si se usan los parámetros `day`, `week`, `month` o `year`)

Ejemplo

En este ejemplo, el comando `start` especifica el 5 de junio de 1999 8:00 am como hora de inicio del primer intervalo que se extraerá. El comando `output` especifica un archivo de salida denominado `myout`.

```
logfile /var/opt/perf/datafiles/logglob
start 6/5/99 8:00 am
output myout
global detail
extract
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -g -b 06/05/99 8:00 AM -f myout -xt
```

stop

Use el comando `stop` para terminar una función `extract` o `export` en la fecha y hora especificadas.

La fecha y la hora de fin predeterminadas son la *última* fecha y hora registrada en el archivo de registro.

Sintaxis

```
start      [date [hora]]
            [today [-día] [hora]]
            [last [-días] [hora]]
            [first [+días] [hora]]
```

Parámetros

<code>date</code>	El formato de <code>date</code> dependerá del idioma original configurado en el sistema. Si el usuario no usa ningún idioma original o ha establecido C como idioma predeterminado, el formato de fecha será <i>mm/dd/aa</i> (mes/día/año), por ejemplo, 09/30/99 en el caso del 30 de septiembre de 1999, para la función <code>extract</code> o <code>export</code> .
<code>time</code>	El formato de <code>time</code> dependerá también del idioma original en uso. En el caso de C, el formato es <i>hh:mm am</i> o <i>hh:mm pm</i> (hora:minuto en formato de 12 horas con el sufijo am/pm). Por ejemplo, 07:00 am son las 7 en punto de la mañana. Se acepta para todos los idiomas la especificación de hora en el formato de 24 horas. Por ejemplo, 23:30 serían las 11.30 pm. Si se introduce la fecha u hora en un formato inaceptable, se mostrará un ejemplo con el formato correcto. Si no se proporciona hora de parada, se propondrá un minuto antes de la medianoche (11:59 pm). La hora de parada de medianoche (12:00 am) de un día dado será el <i>end</i> de ese día (23:59 en un formato de 24 horas).
<code>today</code>	Especifica el día actual. La cualificación del parámetro, como <code>today-days</code> , especifica el número de días con anterioridad a la fecha de hoy. Por ejemplo, <code>today-1</code> indica la fecha de ayer y <code>today-2</code> la fecha de antesdeayer.
<code>last</code>	Puede usarse para representar la última fecha contenida en el archivo de registro. El parámetro <code>last-días</code> especifica el número de días con <i>anterioridad</i> a la última fecha contenida en el archivo de registro.
<code>first</code>	Puede usarse para representar la primera fecha contenida en el archivo de registro. El parámetro <code>first+días</code> especifica el número de días <i>tras</i> la última fecha contenida en el archivo de registro.

Instrucciones sobre su uso

Los siguientes comandos sobrescriben la fecha de fin establecida por el comando `stop`.

- `weekly`
- `monthly`
- `yearly`
- `extract` (si se usan los parámetros `day`, `week`, `month` o `year`)
- `export` (si se usan los parámetros `day`, `week`, `month` o `year`)

Ejemplo

En este ejemplo, el comando `stop` especifica el 5 de junio de 1999 17:00:00 am como hora de fin del último intervalo que se extraerá. El comando `output` especifica un archivo de salida denominado `myout`.

```
extract>
logfile /var/opt/perf/datafiles/logglob
start 6/5/99 8:00 AM
stop 6/5/99 5:00 PM
output myout
global detail
extract
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -g -b 6/5/99 8:00 AM -e 6/5/99 5:00 PM -f myout -xt
```

transaction

Use el comando `transaction` para especificar el tipo de datos de transacción que se extraerán o exportarán.

Sintaxis

```
transaction [on]
               [detail]
               [summary]
               [both]
               [off]
```

Parámetros

<code>on</code>	<code>detail</code>	Consulte “Parámetros” en la descripción del comando application al principio de este capítulo. <code>Summary</code> y <code>both</code> sólo pueden exportarse.
<code>summary</code>		
<code>both</code>		
<code>off</code>		

Ejemplo

El 1 de junio de 1999 se crea un archivo de registro extraído denominado `rxmay99`. Se depura cualquier archivo existente con ese nombre. Se extraen todos los datos del archivo de registro de transacción sin procesar recopilados entre el 1 de mayo de 1999 y el 31 de mayo de 1999.

```
extract>
logfile /var/opt/perf/datafiles/logglob
output rxmay99,purge
global detail
transaction detail
month 9905
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -gt -f rxmay99,purge -xm 9905
```

weekdays

Use el comando `weekdays` para excluir datos de días específicos de la exportación (día 1 = domingo).

Sintaxis

```
weekdays [1|2.....7]
```

Instrucciones sobre su uso

Si desea exportar datos sólo de determinados días de la semana, use este comando para excluir los días de los que *no* desea datos. Los días tienen los siguientes valores:

Domingo	=1
Lunes	=2
Martes	=3
Miércoles	=4
Jueves	=5
Viernes	=6
Sábado	=7

Por ejemplo, si desea exportar sólo los datos registrados del lunes a jueves, *excluya* de la exportación los datos del viernes, sábado y domingo.

Ejemplo

En este ejemplo, cualquier dato global detallado registrado el martes o el jueves se excluye de la exportación. El archivo `output export` contiene las métricas globales especificadas en el archivo de plantilla de exportación `myrept`.

```
extract>  
logfile /var/opt/perf/datafiles/logglob  
global detail  
report myrept  
weekdays 35  
export
```

Use el comando `weekly` para especificar la extracción de datos basada en una semana natural. Una semana se define como siete días que empiezan el lunes y terminan el domingo.

Durante la ejecución, este comando establece las fechas de inicio y parada en las flechas adecuadas, según la semana y el año de los datos extraídos.

Sintaxis

```
weekly    [aass]
           [ss]
```

Parámetros

<code>weekly</code>	Extrae datos de la semana actual (predeterminado).
<code>weekly ss</code>	Extrae datos para una semana específica desde los datos del año actual (donde <i>ss</i> es un número entre 01 y 53).
<code>weekly ssaa</code>	Extrae datos de una semana y año específicos (donde <i>aass</i> es un solo número formado por los últimos dos dígitos del año y los dos dígitos de la semana del año). Por ejemplo, la semana número 20 de 1999 sería <code>weekly 9920</code> .

Si no especifica el archivo de registro antes de ejecutar el comando `weekly`, se usa el archivo `logglob` predeterminado en el directorio `datafiles`.

Instrucciones sobre su uso

Use el comando `weekly` cuando está extrayendo datos para archivarlos semanalmente.

El nombre del archivo de salida está formado por las letras `rxwe` seguidas por los dos dígitos del año y un número de dos dígitos de la semana de la extracción. Por ejemplo, la semana número 12 de 1999 (del lunes 22 de marzo al domingo 29 de marzo) se colocaría en un archivo denominado `rxwe9912`.

El tipo de datos extraídos y resumidos sigue las reglas normales del comando `extract` y puede establecerse antes de ejecutar el comando `weekly`. Se usa esta configuración a menos que ya exista un archivo de salida semanal. En este caso, los datos se anexan a él según el tipo de datos seleccionado originalmente.

El comando `weekly` tiene una función que abre el archivo extraído de la semana *anterior* y comprueba si está lleno; si contiene datos extraídos hasta el último día de la semana. Si no es así, el comando `weekly` anexa datos a este archivo para completar la extracción de la semana anterior.

Por ejemplo, se ejecuta un comando `weekly` el jueves 20 de mayo de 1999. Esto crea un archivo de registro denominado `rxwe199920` que contiene datos desde el 17 de mayo hasta la fecha actual (20 de mayo).

El miércoles 26 de mayo de 1999, se ejecuta otro comando `weekly`. Antes de que se cree el archivo `rxwe199921` para la semana actual, se abre y comprueba el archivo `rxwe199920` de la semana anterior. Cuando se encuentra que está incompleto, se anexan a él los datos para completar la extracción hasta el domingo, mayo 23, 1999. A continuación, se crea el archivo `rxwe199921` para que almacene los datos desde el lunes 24 de mayo de 1999 hasta la fecha actual (26 de mayo).

Mientras se ejecute el comando `weekly` al menos una vez a la semana, esta característica completará cada archivo de la semana antes de crear el archivo de la siguiente semana. Al ver dos archivos semanales adyacentes (por ejemplo, `rxwe199920` y `rxwe199921`), puede suponer que el primer archivo está completo para esa semana y que puede archivarse y depurarse.



Las semanas se enumeran en función del día de inicio. De esta forma, la primera semana del año (semana 01) es la semana que empieza el *primer* lunes de ese año. Cualquier día antes de este lunes pertenece a la última semana del día anterior. Los comandos `weekly` y `extract week` son similares en el sentido de que los dos extraen los datos de una semana natural. El comando `weekly` ignora la configuración del comando `output` y usa en su lugar los nombres de los archivos de salida preferidos. También intenta analizar los datos que faltan al archivo de registro extraído de la semana anterior si aún está presente en el sistema. Por otra parte, el comando `extract week`, usa la configuración del comando `output`. No puede anexas datos al archivo extraído de la semana anterior, ya que no conoce su nombre. El archivo de salida se denomina `rxwe` seguido del año actual (*aaaa*) y la semana del año (*ss*).

Ejemplo

En este ejemplo, el comando `weekly` hace que los datos de la semana actual se extraigan y completa el archivo extraído de la semana anterior, si existe.

```
extract>
logfile /var/opt/perf/datafiles/logglob
global detail
application detail
process detail
weekly
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -gap -xw
```

Use el comando `yearly` para especificar la extracción de datos basada en un año natural.

Durante la ejecución, el comando establece las fechas de inicio y parada en las flechas adecuadas, según el año extraído.

Sintaxis

```
yearly    [aaaa]  
           [aa]
```

Parámetros

<code>yearly</code>	Extrae los datos del año actual (predeterminado).
<code>yearly aa</code>	Extrae los datos de un determinado año (donde <code>aa</code> es un número entre 00 y 99). Las especificaciones 00 a 27 asumen los años 2000 a 2027, mientras que 71 a 99 asumen los años 1971 a 1999.
<code>yearly aaaa</code>	Extrae los datos de un determinado año (donde <code>aaaa</code> es el año completo con el número 1971 a 2027).

Si no especifica el archivo de registro antes de ejecutar el comando `yearly`, se usa el archivo `logglob` predeterminado.

Instrucciones sobre su uso

Use el comando `yearly` cuando está extrayendo datos para archivarlos anualmente.

El nombre del archivo de salida está formado por las letras `rxyr` seguidas por los cuatro dígitos del año de la extracción. De esta forma, los datos de 1999 se colocarían en un archivo denominado `rxyr1999`.

El tipo de datos extraídos y resumidos sigue las reglas normales del comando `extract` y puede establecerse antes de ejecutar el comando `yearly`. Se usa esta configuración a menos que ya exista un archivo de salida anual. En este caso, los datos se anexan a él según el tipo de datos seleccionado originalmente.

El comando `yearly` tiene una función que abre el archivo extraído del año *anterior* y comprueba si está lleno; si contiene datos extraídos hasta el último día del año. Si no es así, el comando `yearly` anexa datos a este archivo para completar la extracción del año anterior.

Por ejemplo, se ejecuta un comando `yearly` el 15 de diciembre de 1998. Esto crea un archivo de registro denominado `rxyr1998` que contiene datos desde el 1 de enero de 1998 hasta la fecha actual (15 de diciembre).

El 5 de enero de 1999, se ejecuta otro comando `yearly`. Antes de que se cree el archivo `rxyr1999` para el año actual, se abre y comprueba el archivo `rxyr1998` del año anterior. Cuando se encuentra que está incompleto, se anexan a él los datos para completar la extracción hasta el 31 de diciembre de 1998. A continuación, se crea el archivo `rxyr1999` para que almacene los datos desde el 1 de enero de 1999 hasta la fecha actual (5 de junio).

Mientras se ejecute el comando `yearly` al menos una vez al año, esta característica completará cada archivo del año antes de crear el archivo del siguiente año. Al ver dos archivos anuales adyacentes (por ejemplo, `rxyr1998` y `rxyr1999`), puede suponer que el primer archivo está completo para ese año y que puede archivar y depurarse.

El párrafo anterior *sólo* es verdadero si los archivos de registro sin procesar tienen un tamaño suficiente para contener *un año completo* de datos. Sería más común asignar un tamaño menor a los archivos del registro sin procesar y ejecutar el comando `yearly` más a menudo (por ejemplo, una vez al mes).



Los comandos `yearly` y `extract year` son similares en el sentido de que los dos extraen los datos de un año natural. El comando `yearly` ignora la configuración del comando `output` y usa en su lugar los nombres de los archivos de salida preferidos. También intenta analizar los datos que faltan al archivo de registro extraído del año anterior si aún está presente en el sistema. Por otra parte, el comando `extract year`, usará la configuración del comando `output`. No puede anexar datos al archivo extraído del año anterior, ya que no conoce su nombre.

Ejemplo

En este ejemplo, los datos de detalle de aplicaciones y globales se anexan al archivo de resumen anual existente (o lo crean, si es necesario). El archivo de salida es `rxyr $aaaa$` (donde $aaaa$ representa el año actual).

```
extract>
logfile /var/opt/perf/datafiles/logglob
global detail
application detail
process off
yearly
```

Para realizar la tarea anterior usando argumentos de línea de comandos, introduzca:

```
extract -ga -xy
```


8 Uso del programa cpsh

El programa cpsh sólo puede usarse si se habilita HP Ops OS Inst to Realtime Inst LTU o Glance Pak Software LTU.

El programa cpsh presenta una nueva interfaz de línea de comandos que permite al usuario ver los datos de métrica en tiempo real recopilados en el sistema monitorizado.

Uso del modo interactivo

El programa cpsh se puede usar en el modo interactivo. Si se ejecuta el comando cpsh sin ninguna opción, el programa cpsh abrirá un nuevo símbolo del sistema. Tras abrirse este símbolo del sistema se pueden realizar diversas tareas para ver los detalles de métrica en tiempo real.

Para abrir el símbolo del sistema cpsh, siga estos pasos:

- 1 Inicie sesión en el sistema (con los privilegios raíz o administrativos) donde se encuentre instalado HP Operations Agent.
- 2 Ejecute el siguiente comando para abrir el símbolo del sistema cpsh del sistema local:

cpsh

Ejecute el siguiente comando para abrir el símbolo del sistema cpsh de un sistema remoto:

cpsh -n <system_name>

donde <system_name> es el nombre de dominio completo del sistema remoto.

o bien

cpsh -n <ip_address>

donde <ip_address> es la dirección IP del sistema remoto.



Al abrir el símbolo del sistema cpsh de un sistema remoto, asegúrese de que el proceso perfd se ejecute en el sistema remoto.

Se abrirá el símbolo del sistema cpsh.



Para ver datos de métrica en un formato de tabla bien estructurado, ejecute el comando cpsh con la opción -t.

Por ejemplo:

cpsh -t

o bien

cpsh -n <system_name> -t

- 3 Para ver los detalles de los comandos disponibles para usar con el símbolo del sistema `cpsh`, escriba **help**.

Ver métrica en tiempo real

El símbolo del sistema `cpsh` permite ver los valores en tiempo real de la métrica disponible. Antes de realizar una operación con el símbolo `cpsh`, se deberá establecer el contexto de métrica. El demonio `perfd` y las utilidades asociadas procesan los datos disponibles en función de las clases de métrica. Por lo tanto, cuando desee usar la utilidad `cpsh` para ver datos en tiempo real será necesario establecer la clase de métrica antes de realizar ninguna operación para ver los datos disponibles.

Para ver los valores de métrica en tiempo real de una clase de métrica, siga los pasos siguientes:

- 1 Tras mostrarse el símbolo del sistema `cpsh`, escriba **class** `<metric_class>`.
- 2 Para obtener una lista de la métrica actualmente establecida para una clase en particular, escriba **list**. Se mostrará una lista de toda la métrica predeterminada de una clase métrica especificada.
- 3 Para ver los valores de métrica que pertenecen a la clase especificada, escriba **push** al mostrarse el símbolo del sistema `cpsh`. El programa `cpsh` muestra los valores de la métrica en tiempo real en un formato tabular.
- 4 Para regresar al símbolo del sistema `cpsh`, pulse **Ctrl+C**.

Modificar una clase de métrica

Es posible agregar métrica disponible adicional a la lista de métrica predeterminada para una clase de métrica. Para agregar o eliminar métrica de una clase de métrica en el símbolo del sistema `cpsh`, siga los pasos siguientes:

- 1 Abra el símbolo del sistema `cpsh`.
- 2 Tras mostrarse el símbolo del sistema `cpsh`, escriba **class** `<metric_class>`.
- 3 Escriba **list**. Se mostrará una lista de toda la métrica predeterminada de una clase métrica especificada.
- 4 Para eliminar una métrica, siga estos pasos:
 - a Tras mostrarse el símbolo del sistema `cpsh`, escriba **delete** `<metric_name>`.
 - b Escriba **list**. La lista de métrica para la clase de métrica especificada no incluirá la métrica eliminada.
- 5 Para agregar una métrica a la clase de métrica, siga los pasos siguientes:
 - a Tras mostrarse el símbolo del sistema `cpsh`, escriba **add** `<metric_name>`.
 - b Escriba **list**. La lista de métrica para la clase de métrica especificada incluirá la métrica recién agregada.

Ver toda la métrica disponible

Para ver toda la métrica disponible que pertenece a una clase de métrica, siga los pasos siguientes:

- 1 Abra el símbolo del sistema `cps`.
- 2 Tras mostrarse el símbolo del sistema `cps`, escriba **class** `<metric_class>`.
- 3 Escriba **list all**. Se mostrará una lista de toda la métrica disponible que pertenece a la clase métrica especificada.

Organizar una clase de métrica

Se puede reorganizar una clase de métrica sin realizar operaciones secuenciales de agregar y eliminar en la clase. Para reorganizar una clase de métrica con objeto de incluir una métrica en particular, siga los pasos siguientes:

- 1 Abra el símbolo del sistema `cps`.
- 2 Tras mostrarse el símbolo del sistema `cps`, escriba **class** `<metric_class>`.
- 3 Escriba **init** `<metric_name>` `<metric_name>` `<metric_name>`

La clase de métrica especificada sólo incorporará la métrica especificada con el comando `init`.

Ver Ayuda de métrica

El símbolo de sistema `cps` permite ver la descripción de la métrica en tiempo real. Para ver la descripción de la métrica, siga estos pasos:

- 1 Abra el símbolo del sistema `cps`.
- 2 Escriba **class** `<metric_class>`.
- 3 Tras mostrarse el símbolo del sistema `cps`, escriba **help** `<metric_name>`. Se mostrará una descripción de detalles de la métrica.

Ver datos de métrica resumidos

El símbolo del sistema `cps` permite ver datos resumidos respecto a la métrica de las clases `GLOBAL` y `TABLE`. Para ver los datos resumidos, siga estos pasos:

- 1 Abra el símbolo del sistema `cps`.
- 2 Tras mostrarse el símbolo del sistema `cps`, escriba **class gbl** o **class tbl**.
- 3 Escriba **summ** `<interval>`. En este caso, `<interval>` es el intervalo de resumen especificado en segundos. `<interval>` debe ser un múltiplo de intervalo de recopilación del servidor `perf` al que está conectado `cps`.

La utilidad `cps` muestra las siguientes medidas de los valores de métrica que pertenecen a la clase de métrica seleccionada:

- Máximo
- Mínimo
- Promedio
- Desviación estándar

9 Alarmas de rendimiento

Componente Performance Collection permite al usuario definir alarmas. Estas alarmas notifican al usuario si `scope` o la métrica DSI cumple o excede las condiciones establecidas.

Para definir alarmas debe especificar condiciones en todos los sistemas monitorizados. Cuando éstas se cumplan, activarán una alerta o acción. Las alarmas pueden ser definidas en el archivo de texto de definiciones de alarma `alarmdef`.

A medida que `scope` u otros colectores registran datos, éstos son comparados con las definiciones de alarma si se cumple una condición. Cuando ello ocurra, se activará una acción o alerta.

Con el generador de alarmas en tiempo real, el usuario podrá realizar las tareas siguientes:

- Enviar notificaciones de alerta a la consola HPOM.
- Crear una captura SNMP cuando se genere una notificación de alerta.
- Reenviar la captura SNMP a un escucha de alertas SNMP.
- Realizar acciones locales en los sistemas monitorizados.

El usuario podrá analizar datos históricos del archivo de registro con las definiciones de alarma y notificar los resultados con el comando `analyze` del programa `utility`.

Procesamiento de alarmas

A medida que Componente Performance Collection recopila datos, los datos recopilados son comparados con las condiciones de alarma definidas en el archivo `alarmdef` para determinar si se han cumplido las condiciones. Cuando se cumple una condición se genera una alarma y se realizan las acciones vinculadas con las alarmas (`ALERT`, `PRINT` y `EXEC`).

A medida que se recopilan datos en los archivos de registro, éstos son comparados con las definiciones de alarma en el archivo `alarmdef`. Cuando se cumple una condición, las acciones definidas en la definición de alarma son efectuadas. No obstante, si los datos no están registrados en los archivos de registro (por ejemplo, si se han establecido valores demasiado altos para los parámetros de umbral), las alarmas no serán generadas ni siquiera si se cumplen las condiciones de alarma en el archivo `alarmdef`. Consulte [Thresholds](#) en la página 25 para obtener los valores de umbral de las distintas clases de métrica.

Entre las acciones definidas en la definición de alarma se encuentran:

- acciones locales realizadas con los comandos UNIX.
- mensajes enviados a Network Node Manager y a Operations Manager.

Generador de alarmas

El generador de alarmas de Componente Performance Collection es responsable de la comunicación de notificaciones de alarma. El generador de alarmas consta del servidor del generador de alarmas (`perfalarm`), la base de datos del generador de alarmas (`agdb`) y el programa de utilidades `agsysdb`.

El programa `agdb` contiene una lista de los nodos SNMP. El programa `agsysdb` permite mostrar y cambiar las acciones que propician los eventos de alarma.

Al iniciar Componente Performance Collection se inicia `perfalarm`, el cual lee `agdb` para determinar dónde enviar notificaciones de alarma y si es necesario hacerlo. Comprueba también si el sistema contiene un agente de Operations Manager.

Use la opción de línea de comando siguiente para obtener una lista de las ubicaciones donde son enviadas las notificaciones de alerta:

```
agsysdb -l
```

Envío de capturas SNMP a Network Node Manager

Para enviar capturas SNMP a Network Node Manager, deberá agregar el nombre de su sistema en `agdb` de Componente Performance Collection con el comando siguiente:

```
agsysdb -add nombre del sistema
```

Cada ALERTA que se genere hará que se envíe una captura SNMP al sistema definido por el usuario. El texto de la captura contendrá el mismo mensaje que la ALERTA.

Para detener el envío de capturas SNMP al sistema, elimine el nombre del sistema de `agdb` con el comando siguiente:

```
agsysdb -delete nombre del sistema
```

Envío de mensajes a Operations Manager

Las notificaciones de alerta podrán ser enviadas a Operations Manager si hay un agente de Operations Manager en el mismo sistema que Componente Performance Collection. El agente de Operations Manager se comunica con el sistema central de Operations Manager.

Por defecto, si el agente de Operations Manager está siendo ejecutado en el sistema de Componente Performance Collection, el generador de alarmas *no* ejecutará acciones locales que hayan sido definidas en las alarmas de la instrucción EXEC. En cambio, enviará un mensaje al explorador de eventos de Operations Manager. Si el agente de Operations Manager *no* está siendo ejecutado en el sistema de Componente Performance Collection, el generador de alarmas no hará ningún intento de envío de notificaciones de alerta a Operations Manager y se ejecutarán las acciones locales.

Con el siguiente comando el usuario podrá cambiar el valor predeterminado para detener el envío de información a Operations Manager aunque se esté siendo ejecutando un agente de Operations Manager en el sistema de Componente Performance Collection:

```
agsysdb -ovo OFF
```

Para enviar capturas de Componente Performance Collection a otro nodo, agregue las entradas siguientes al archivo `/etc/services`.

```
snmp-trap 162/tcp # SNMPTRAP  
snmp-trap 162/udp # SNMPTRAP
```

En este caso, 162 especifica el número de puerto. Si desea que Componente Performance Collection envíe capturas a otro modo, comprobará el archivo `/etc/services` para averiguar si contiene la cadena de capturas snmp. Si no está disponible esta entrada, las capturas no serán enviadas al otro nodo.

Ejecución de acciones locales

Las acciones locales en instrucciones EXEC serán ejecutadas sin que se ejecute un agente de Operations Manager en el sistema de Componente Performance Collection.

El valor predeterminado para desactivar las acciones locales puede ser cambiado como sigue:

```
agsysdb -actions off
```

Si desea que las acciones locales sean *siempre* ejecutadas, incluso cuando se está ejecutando un agente de Operations Manager, escriba lo siguiente:

```
agsysdb -actions always
```

La siguiente tabla incluye las configuraciones para enviar información a Operations Manager y ejecutar acciones locales:

Tabla 10 Configuración para enviar información a Operations Manager y ejecutar acciones locales

Indicadores	Agente de Operations Manager en ejecución	Sin agente de Operations Manager en ejecución
Operations Manager Indicador		
off	No se produce el envío de notificaciones de alerta a Operations Manager.	No se produce el envío de notificaciones de alerta a Operations Manager.
on	Se produce el envío de notificaciones de alerta a Operations Manager.	Se produce el envío de notificaciones de alerta a Operations Manager.
Indicador de acciones locales		
off	Sin acciones locales ejecutadas.	Sin acciones locales ejecutadas.
always	Las acciones locales son ejecutadas incluso con agente de Operations Manager en ejecución.	Acciones locales ejecutadas.
on	Las acciones locales son enviadas a Operations Manager.	Acciones locales ejecutadas.

Errores de procesamiento de alarmas

El último error producido durante el envío de una alarma queda registrado en agdb. Para ver el contenido de agdb, escriba:

```
agsysdb -1
```

Se muestra la siguiente información:

```
PA alarming status:

OVO messages : on      Last Error : none
Exec Actions  : on
Analysis system: <hostname>, Key=<ip address>
PerfView      : no Last Error : <error number>
SNMP          : yes Last Error : <error number>
```

Análisis de datos históricos de alarmas

El comando `analyze` del programa `utility` permite localizar condiciones de alarma en los datos del archivo de registro (consulte el [Capítulo 5, Comandos del programa utility](#)). Esto difiere del procesamiento de alarmas en tiempo real explicado anteriormente ya que en este caso se comparan datos históricos del archivo de registro con las definiciones de alarma del archivo `alarmdef` para determinar qué condiciones de alarma han sido activadas.

Ejemplos de información de alarma en datos históricos

Los siguientes ejemplos muestran lo que es notificado al analizar las condiciones de alarma de datos históricos.

En el caso del primer ejemplo se han definido en la definición de alarma las instrucciones `START`, `END` y `REPEAT`. Se enunciará un evento de inicio de alarma cada vez que una alarma cumpla todas las condiciones correspondientes a la duración especificada. Cuando dejen de cumplirse estas condiciones, se enunciará un evento de fin de alarma. Si se cumple una condición de alarma durante un periodo de tiempo lo suficientemente largo como para que se genere otra alarma sin que ésta haya finalizado con anterioridad, se enunciará un evento de repetición de alarma.

Cada evento enunciado muestra la fecha y hora, número de alarma y evento de alarma. Las acciones `EXEC` *no* serán ejecutadas sino enunciadas con las sustituciones de parámetros requeridos existentes.

```
05/10/99 11:15 ALARM [1] START
CRITICAL: CPU test 99.97%
05/10/99 11:20 ALARM [1] REPEAT
WARNING: CPU test 99.997%
05/10/99 11:25 ALARM [1] END
RESET: CPU test 22.86%
EXEC: end.script
```

Si usa una estación de trabajo a color, se resaltará la salida siguiente:

```
CRITICAL statements are RED
MAJOR statements are MAGENTA
MINOR statements are YELLOW
WARNING statements are CYAN
NORMAL statements are GREEN
```

El siguiente ejemplo muestra el resumen de alarma que se obtiene después de que sean enunciados los eventos de alarma. La primera columna incluye el número de alarma, la segunda el número de veces que se produce una condición de alarma y la tercera la duración total de la condición de alarma.

Performance Alarm Summary:

Alarm	Count	Minutes
1	574	2865
2	0	0

```
Analysis coverage using "alarmdef":
Start: 05/04/99 08:00      Stop: 05/06/99 23:59
Total time analyzed:  Days: 2  Hours: 15  Minutes: 59
```

Componentes de definiciones de alarma

Las alarmas ocurren cuando una o más de las condiciones definidas por el usuario continúan válidas durante un periodo de tiempo determinado. La definición de alarma puede incluir una acción que será realizada al inicio o fin de la alarma.

Una condición es una comparación entre dos o más ítems. Los ítems que se comparan pueden ser nombres de métrica, constantes o variables. Por ejemplo:

```
ALARM gbl_cpu_total_util > 95 FOR 5 MINUTES
```

Puede especificarse que la acción sea realizada cuando se inicie, finalice o se repita la alarma. La acción puede ser una de las siguientes:

- una acción **ALERT** que envía un mensaje a Operations Manager o una captura SNMP a Network Node Manager
- una acción **EXEC** que ejecuta un comando UNIX, o
- una acción **PRINT** que envía un mensaje a stdout al ser procesado con el programa utility.

Por ejemplo:

```
ALARM gbl_swap_space_util > 95 FOR 5 MINUTES
START
  RED ALERT "Global swap space is nearly full"
END
  RESET ALERT "End of global swap space full condition"
```

Se pueden crear acciones más complejas con lógica booleana, bucles en datos de múltiples instancias como aplicaciones, y variables. (Para mayor información, consulte la sección siguiente [Referencia de sintaxis de alarma](#)).

La instrucción INCLUDE permite también identificar archivos de definición de alarmas adicionales que desee utilizar. Por ejemplo, para dividir sus definiciones de alarma en archivos más pequeños.

Referencia de sintaxis de alarma

Esta sección describe las instrucciones disponibles en la sintaxis de alarma. Es aconsejable consultar el archivo `alarmdef` para ver ejemplos sobre cómo usar la sintaxis para crear definiciones de alarma útiles.

Sintaxis de alarma

```
ALARM condition [[AND,OR]condition]
  FOR duration [SECONDS, MINUTES]
  [TYPE="string"]
  [SERVICE="string"]
  [SEVERITY=integer]
  [START action]
  [REPEAT EVERY duration [SECONDS, MINUTES] action]
  [END action]
[RED, CRITICAL, ORANGE, MAJOR, YELLOW, MINOR, CYAN, WARNING,
 GREEN, NORMAL, RESET] ALERT message
EXEC "UNIX command"
PRINT message
IF condition
  THEN action
  [ELSE action]
{APPLICATION, PROCESS, DISK, LVOLUME, TRANSACTION, NETIF, CPU,
 FILESYSTEM} LOOP action
INCLUDE "filename"
USE "data source name"
[VAR] name = value
ALIAS name = "replaced-name"
SYMPTOM variable [ TYPE = {CPU, DISK, MEMORY, NETWORK}]
  RULE condition PROB probability
  [RULE condition PROB probability]
.
.
```

Convenciones

- Las llaves de apertura y cierre ({ }) indican que una de las opciones es requerida.
- Los corchetes ([]) indican un ítem opcional.
- ítems separados por comas entre corchetes o llaves son opciones. Seleccione sólo uno.
- Las cursivas indican un nombre de variable que el usuario sustituirá.
- Todas las palabras clave de sintaxis están en mayúscula.

Elementos comunes

Los elementos que se describen a continuación son utilizados en diversas instrucciones de sintaxis de alarma.

- comentarios
- instrucciones compuestas
- condiciones
- constantes
- expresiones
- nombres de métrica
- mensajes

Comentarios

Los comentarios pueden estar precedidos por bien las barras diagonales (//) o el signo de almohadilla (#). En ambos casos el comentario terminará al final de la línea. Por ejemplo:

```
# cualquier texto o caracteres
o
// cualquier texto o caracteres
```

Instrucciones compuestas

Las instrucciones compuestas permiten que una lista de instrucciones sea ejecutada como una instrucción única. Una instrucción compuesta es una lista de instrucciones entre llaves ({}). Use la instrucción compuesta con las instrucciones IF y LOOP, y las cláusulas START, REPEAT y END de la instrucción ALARM. Las instrucciones compuestas no pueden incluir las instrucciones ALARM y SYMPTOM.

```
{
  statement
  statement
}
```

En el ejemplo anterior `highest_cpu` define una variable. El valor `highest_cpu` es guardado y sólo notifica al usuario si dicho valor `highest_cpu` es superado por un valor `highest_cpu` más alto.

```
highest_cpu = highest_cpu
IF gbl_cpu_total_util > highest_cpu THEN
  // Begin compound statement
  {
    highest_cpu = gbl_cpu_total_util
    ALERT "Our new high CPU value is ", highest_cpu, "%"
  }
  // End compound statement
```

Condiciones

Una condición puede definirse como una comparación entre dos ítems.

```
item1 {>, <, >=, <=, ==, !=}item2  
[AND, OR[item3 {>, <, >=, <=, ==, !=}item4]]
```

en donde "==" significa "igual" y "!=" significa "no igual".

Las condiciones son utilizadas en las instrucciones ALARM, IF y SYMPTOM. Un ítem puede ser un nombre de métrica, una constante numérica, una cadena alfanumérica entre comillas, un alias o una variable. Cuando se comparan ítems alfanuméricos, sólo pueden utilizarse como operadores == o !=.

Constantes

Las constantes pueden ser tanto alfanuméricas como numéricas. Una constante alfanumérica debe estar entre comillas dobles. Por ejemplo:

```
345  
345.2  
"Time is"
```

Las constantes son útiles en expresiones y condiciones. Por ejemplo, puede que se desee comparar una métrica con un valor numérico constante en una condición para generar una alarma si el valor es demasiado alto, como:

```
gbl_cpu_total_util > 95
```

Expresiones

Las expresiones aritméticas efectúan una o más operaciones aritméticas en dos o más operandos. Se podrán usar expresiones en cualquier lugar donde se usan valores numéricos. Los operadores aritméticos válidos son:

```
+, -, *, /
```

Los paréntesis permiten controlar qué partes de una expresión son evaluadas en primer lugar.

Por ejemplo:

```
Iteration + 1  
gbl_cpu_total_util - gbl_cpu_user_mode_util  
( 100 - gbl_cpu_total_util ) / 100.0
```

Nombres de métrica

Cuando se especifica un nombre de métrica en la definición de alarma, el valor actual de la métrica es sustituido. Los nombres de métrica deberán escribirse exactamente igual que aparecen en la definición de métrica, excepto en el caso de que se haga distinción entre mayúsculas y minúsculas. Las definiciones de métrica se encuentran en el *Dictionary of Operating Systems Performance Metrics* de Componente Performance Collection.

Se recomienda el uso de nombres de métrica completos si la métrica procede de un origen de datos distinto al origen de datos SCOPE (como por ejemplo métrica DSI).

El formato para especificar una métrica completa es el siguiente:

```
data_source:instance(class):metric_name
```


Una métrica global en el origen de datos SCOPE no requerirá el nombre completo. Por ejemplo:

```
metric_1
```

Una métrica de aplicación que esté disponible para todas las aplicaciones definidas en el origen de datos SCOPE, requerirá el nombre de aplicación. Por ejemplo:

```
application_1:metric_1
```

Para tipos de datos de instancias múltiples, como `application`, `process`, `disk`, `netif`, `transaction`, `lvolume`, `cpu` y `filesystem`, deberá asociarse la métrica con el nombre del tipo de datos, excepto si se usa la instrucción `LOOP`. Para ello, especifíquese el nombre del tipo de aplicación seguido de una coma, y a continuación el nombre de métrica. Por ejemplo, `other_apps:app_cpu_total_util` especifica el uso total de CPU de la aplicación `other_apps`.



Cuando se especifiquen métricas de instancias múltiples completas y se usen alias con alias, si uno de los alias tiene un identificador de clase, se recomienda que se use la sintaxis que se muestra en el ejemplo siguiente:

```
alias my_fs="/dev/vg01/lvol1(LVOLUME)"  
alarm my_fs:LV_SPACE_UTIL > 50 for 5 minutes
```

Si usa un nombre de aplicación que contiene un espacio insertado, deberá sustituir el espacio por un guión bajo (`_`). Por ejemplo, `application 1` deberá ser sustituido por `application_1`. Para mayor información sobre el uso de nombres que contengan caracteres especiales, o nombres en los que se haga distinción entre mayúscula y minúscula, consulte la [Instrucción ALIAS](#) en la página 182.

Si tiene un disco llamado “other” y una aplicación llamada “other” también, posiblemente tenga que especificar tanto la clase como la instancia:

```
other (disk):metric_1
```

Una métrica global de un archivo de registro extraído (en donde `scope_extract` es el nombre de origen de datos) quedaría especificada del siguiente modo:

```
scope_extract:application_1:metric_1
```

Una métrica DSI quedaría especificada del siguiente modo:

```
dsi_data_source:dsi_class:metric_name
```



Todo nombre de métrica que contenga caracteres especiales (por ejemplo, asteriscos) deberá tener alias antes de ser especificado.

Mensajes

Un mensaje es la información que es enviada a la instrucción `PRINT` o `ALERT`. Puede constar de cualquier combinación de cadenas alfanuméricas entre comillas, constantes numéricas, expresiones y variables. Los elementos que contiene el mensaje están separados por comas. Por ejemplo:

```
RED ALERT "cpu utilization=", gbl_cpu_total_util
```

Las constantes numéricas, métricas y expresiones pueden ser formateadas en cuanto al ancho y número de decimales. *Width* especifica el ancho del campo; *decimals* especifica el número de decimales que se va a usar. Los valores numéricos están alineados a la derecha. El signo - (signo menos) especifica la alineación a la izquierda. Las cadenas alfanuméricas siempre están alineadas a la izquierda. Por ejemplo:

```
metric names [|[-]width[|decimals]]
gbl_cpu_total_util|6|2   formats as '100.00'
(100.32 + 20)|6         formats as '  120'
gbl_cpu_total_util|-6|0  formats as '100  '
gbl_cpu_total_util|10|2  formats as '  99.13'
gbl_cpu_total_util|10|4  formats as '  99.1300'
```

Instrucción ALARM

La instrucción ALARM define una condición o conjunto de condiciones y la duración para que las condiciones sean true. Dentro de la instrucción ALARM se pueden definir acciones que se efectúen cuando se inicie, repita o termine la condición de alarma. Entre las condiciones o eventos que se pueden definir como alarmas, cabe citar:

- cuando el espacio swap total haya estado casi lleno durante 5 minutos.
- cuando la frecuencia de paginación de memoria haya sido demasiado alta durante 1 intervalo.
- cuando el uso del CPU en ejecución sea del 75% durante los últimos diez minutos.

Sintaxis

```
ALARM condition [[AND,OR]condition]
FOR duration{SECONDS, MINUTES}
[TYPE="string"]
[SERVICE="string"]
[SEVERITY=integer]
[START action]
[REPEAT EVERY duration {SECONDS, MINUTES} action]
[END action]
```

- La instrucción **ALARM** debe ser una instrucción de nivel superior. No puede ser anidada en ninguna otra instrucción. No obstante, se pueden incluir varias condiciones ALARM en una misma instrucción ALARM. Si las condiciones están vinculadas por **AND**, todas las condiciones deberán ser true para que la alarma se active. Si están vinculadas por **OR**, cualquiera de las condiciones activará la alarma.
- **TYPE** es una cadena entre comillas que puede contener hasta 38 caracteres. Cuando envíe alarmas, use **TYPE** para categorizar alarmas y especificar el nombre de la plantilla de gráficos que desea usar.
- **SERVICE** es una cadena entre comillas que puede contener hasta 200 caracteres. Si usa ServiceNavigator, cuya última versión incorpora el ITO 5.0, podrá vincular las alarmas de Componente Performance Collection con los servicios que haya definido en ServiceNavigator (consulte la *Operations ServiceNavigator Concepts and Configuration Guide* de HP).

```
SERVICE="Service_id"
```

- **SEVERITY** es un entero de 0 a 32767.

- **START, REPEAT y END** son *palabras clave* que permiten especificar la acción a realizar cuando se cumplen, se vuelven a cumplir o se dejan de cumplir las condiciones de alarma. Las instrucciones ALARM deberán contener al menos una de las palabras clave **START, REPEAT o END**. Cada una de estas palabra clave está seguida de una *action*.
- *action* – La acción que se usa más frecuentemente con ALARM START, REPEAT o END es la instrucción ALERT. No obstante, también se puede usar la instrucción EXEC para enviar un mensaje o para ejecutar un archivo por lotes, o una instrucción PRINT en el caso de analizar archivos de registro históricos con el programa *utility*. Es válida cualquier instrucción de sintaxis excepto otra ALARM.

Las acciones START, REPEAT y END pueden ser instrucciones compuestas. Por ejemplo, podrá usar instrucciones compuestas tanto con ALERT como con EXEC.

- *Conditions* – Una condición puede definirse como una comparación entre dos ítems.

```
item1 {>, <, >=, <=, ==, !=}item2
  [AND, OR[item3 {>, <, >=, <=, ==, !=}item4]]
```

en donde "==" significa "igual" y "!=" significa "no igual".

Un ítem puede ser un nombre de métrica, una constante numérica, una cadena alfanumérica entre comillas, un alias o una variable. Cuando se comparan ítems alfanuméricos, sólo pueden utilizarse como operadores == o !=.

Se pueden usar condiciones compuestas especificando los operadores “OR” o “AND” entre subcondiciones. Por ejemplo:

```
ALARM gbl_cpu_total_util > 90 AND
gbl_pri_queue > 1 for 5 minutes
```

- También se pueden usar condiciones compuestas *sin* especificar los operadores “OR” o “AND” entre subcondiciones. Por ejemplo:

```
ALARM gbl_cpu_total_util > 90
gbl_cpu_sys_mode_util > 50 for 5 minutes
```

causará una alarma cuando ambas condiciones sean true.

FOR *duration* **SECONDS, MINUTES** especifica el periodo de tiempo durante el cual la condición debe ser true para que la alarma se active.

Tenga cuidado al especificar duraciones de menos de un minuto, en especial cuando haya múltiples orígenes de datos en el sistema. El rendimiento puede verse seriamente afectado si cada origen de datos tiene que ser sondeado en busca de datos en intervalos muy cortos. La duración deberá ser un múltiplo del intervalo de recolección más largo de la métrica mencionada en la condición de alarma.

Para datos de *scope*, la duración es de cinco minutos, mientras que la duración para datos de proceso es de un minuto. En el caso de datos DSI, la duración es de cinco segundos o más.

- **REPEAT EVERY** *duration* **SECONDS, MINUTES** especifica el periodo de tiempo que transcurre para que se repita la alarma.

Modo en que es utilizado

El ciclo de alarma comienza en el primer intervalo en el que todas las condiciones de alarma con el operador AND o una de las condiciones con OR son true durante al menos la duración especificada. En dicho momento el generador de alarmas ejecuta una *acción* START, y en cada intervalo siguiente comprueba la condición REPEAT. Si ha transcurrido el tiempo suficiente, se ejecuta la *acción* correspondiente a la cláusula REPEAT. (Este proceso continua hasta que una o más condiciones de alarma sea false). Con ello finaliza el ciclo de alarma y la instrucción END es ejecutada si hay una condición false.

Para recibir notificación de alarma, use la instrucción ALERT en las instrucciones START y END. Si no especifica END ALERT, el generador de alarmas enviará automáticamente una alarma a Operations Manager y una captura SNMP a Network Node Manager.

Ejemplos

El siguiente ejemplo de ALARM envía una alerta roja cuando la utilización swap es alta durante 5 minutos. Es similar a una condición de alarma en el archivo predeterminado alarmdef. No lo agregue a su archivo alarmdef sin haber eliminado antes la condición de alarma predeterminada, o los mensajes de alerta subsiguientes podrán ser confusos.

```
ALARM gbl_swap_space_util > 90 FOR 5 MINUTES
  START
    RED ALERT "swap utilization is very high "
  REPEAT EVERY 15 MINUTES
    RED ALERT "swap utilization is still very high "
  END
  RESET ALERT "End of swap utilization condition"
```

El anterior ejemplo de ALARM comprueba si la métrica gbl_swap_space_util es superior a 90. ALERT será enviada a Network Node Manager mediante una captura SNMP o como mensaje de Operations Manager dependiendo del modo en que haya configurado el generador de alarmas.

La instrucción REPEAT comprueba la condición gbl_swap_space_util cada 15 segundos. La instrucción REPEAT enviará el mensaje “swap utilization is still very high” cada 15 minutos siempre y cuando la métrica sea superior a 90.

Si la condición gbl_swap_space_util es inferior a 90, se enviará la instrucción RESET ALERT con el mensaje “End of swap utilization condition”.

El ejemplo siguiente define una acción compuesta en la instrucción ALARM. El siguiente ejemplo muestra cómo hacer que se envíe un mensaje cuando ocurre un evento.

```
ALARM gbl_cpu_total_util > 90 FOR 5 MINUTES
  START
  {
    RED ALERT "Your CPU is busy."
    EXEC "echo 'cpu is too high'| mailx root"
  }
  END
  RESET ALERT "CPU no longer busy."
```

ALERT puede activar el envío de una captura SNMP a Network Node Manager o de un mensaje a Operations Manager. EXEC activará el envío de un mensaje de correo como acción local en su sistema de Componente Performance Collection dependiendo del modo en que haya configurado el generador de alarmas.

Por defecto, si el agente de Operations Manager está en ejecución, la acción local no será ejecutada. En cambio, será enviada como mensaje a Operations Manager.

Los dos ejemplos siguientes muestran el uso de condiciones múltiples. Es posible tener más de una condición de prueba en la instrucción ALARM. En dicho caso, todas las instrucciones deberán ser true para que se envíe la ALERT.

El siguiente ejemplo de ALARM comprueba la métrica `gbl_cpu_total_util` y `gbl_cpu_sys_mode_util`. Si ambas condiciones son true, la instrucción RED ALERT envía una alerta roja. Si una de las condiciones de prueba se vuelve false, se enviará RESET.

```
ALARM gbl_cpu_total_util > 90
      AND gbl_cpu_sys_mode_util > 50 FOR 5 MINUTES
START
  RED ALERT "CPU busy and Sys Mode CPU util is high."
END
  RESET ALERT "The CPU alert is now over."
```

El siguiente ejemplo de ALARM comprueba la métrica `gbl_cpu_total_util` y `gbl_cpu_sys_mode_util`. Si alguna de las condiciones es true, la instrucción RED ALERT enviará una alerta roja.

```
ALARM gbl_cpu_total_util > 90
      OR
      gbl_cpu_sys_mode_util > 50 FOR 10 MINUTES
START
  RED ALERT "Either total CPU util or sys mode CPU high"
```



No use métrica que esté registrada en intervalos diferentes en la misma alarma. Por ejemplo, no itere un proceso (registrado en intervalos de 1 minuto) basado en el valor de una métrica global (registrado en intervalos de 5 minutos) en una instrucción como la que sigue:

```
IF global_metric THEN
  PROCESS LOOP...
```

Los diferentes intervalos no pueden ser sincronizados como se espera, de modo que los resultados no serán válidos.



En el caso de GlancePlus, use la métrica de proceso dentro de un bucle de proceso para enviar alarmas a todos los procesos.

Instrucción ALERT

La instrucción ALERT permite que se envíe un mensaje a Network Node Manager o Operations Manager. La instrucción ALERT se usa frecuentemente como una acción de una ALARM. También puede ser utilizada en una instrucción IF para enviar un mensaje tan pronto como se detecte una condición o después de que la duración concluya. Si se usa una ALERT fuera de una instrucción ALARM o IF, el mensaje será enviado en cada intervalo.

Sintaxis

```
[RED, CRITICAL, ORANGE, MAJOR, YELLOW, MINOR, CYAN, WARNING, GREEN, NORMAL, RESET] ALERT mensaje
```

- **RED** es sinónimo de **CRITICAL**, **ORANGE** es sinónimo de **MAJOR**, **YELLOW** es sinónimo de **MINOR**, **CYAN** es sinónimo de **WARNING** y **GREEN** es sinónimo de **NORMAL**. Estas palabras clave cambian el símbolo de alarma al color asociado con la condición de alarma.
- **RESET** — Envía una **RESET ALERT** con un mensaje cuando concluye la condición ALARM. Si no ha definido ninguna RESET ALERT en la definición de alarma, enviará una **RESET ALERT** sin mensaje cuando concluya la condición ALARM.
- *message* — Una combinación de cadenas y valores numéricos usados para crear un mensaje. Los valores numéricos pueden ser formateados con los parámetros `[[-]width [decimals]]`. *Width* especifica el ancho del campo; *decimals* especifica el número de decimales que se va a usar. Los valores numéricos están alineados a la derecha. El signo - (signo menos) especifica la alineación a la izquierda. Las cadenas alfanuméricas siempre están alineadas a la izquierda.

Modo en que es utilizado

ALERT activará asimismo el envío de una captura SNMP a Network Node Manager o de un mensaje a Operations Manager dependiendo del modo en que haya configurado el generador de alarmas. En el caso de mensajes de alerta enviados a Operations Manager, las WARNINGS deberán ser mostradas en azul en el explorador de mensajes.

Ejemplo

Una instrucción ALERT común es:

```
RED ALERT "CPU utilization = ", gbl_cpu_total_util
```

Si dispone de Network Node Manager, esta instrucción creará una alarma de gravedad crítica en la ventana del Explorador de alarmas de Network Node Manager.

Instrucción EXEC

La instrucción EXEC permite especificar el comando de sistema (UNIX o Windows) que será ejecutado en el sistema local. Por ejemplo, se puede usar la instrucción EXEC para enviar correo al administrador de TI cada vez que se cumpla una condición.

EXEC deberá ser utilizado dentro de una instrucción ALARM o IF de manera que el comando sólo sea ejecutado cuando se cumplan las condiciones establecidas. Si se usa la instrucción EXEC fuera de la instrucción ALARM o IF, la acción será realizada en intervalos impredecibles.

Sintaxis

EXEC *"system command"*

system command — un comando que será ejecutado en el sistema local.

No use comillas dobles (") en las instrucciones EXEC. El hacerlo puede ocasionar que perfalarm no envíe la alarma a HPOM. Use en su lugar las comillas simples ('). Por ejemplo:

```
EXEC "echo 'performance problem detected' "
```

```
EXEC "mkdir c:\directory\filename"
```

La sintaxis de la instrucción EXEC requiere el nombre de ruta del archivo que quedará entre comillas. No obstante, si el nombre de ruta contiene espacios, el nombre de ruta deberá estar entre comillas simples, que a su vez estarán dentro de comillas dobles.

Ejemplo:

```
EXEC "'C:\Program Files\Mail Program\SendMail.exe'"
```

Si algún argumento del comando del sistema de la instrucción EXEC contiene comillas simples, el programa deberá quedar entre comillas simples puesto que el primer par de comillas simples (') cambiará a comillas dobles (") al ejecutar usted el comando con la instrucción EXEC.

Ejemplo:

```
EXEC "'echo' 'test execution'"
```

En el ejemplo anterior, echo es el programa que queda dentro de las comillas simples puesto que contiene un argumento (en este caso, test execution) entre comillas simples. Además, según la sintaxis de la instrucción EXEC, toda la cadena del comando deberá estar entre comillas dobles.

No use comillas dobles (") en instrucciones EXEC; perfalarm no enviará correctamente la alarma a HPOM. Use en su lugar las comillas simples (').

Por ejemplo:

```
EXEC "'echo' 'dialog performance problem'"
```

En el ejemplo anterior, echo es el programa que queda dentro de las comillas simples puesto que contiene un argumento (en este caso, dialog performance problem) entre comillas simples. Además, según la sintaxis de la instrucción EXEC, toda la cadena del comando deberá estar entre comillas dobles.

Modo en que es utilizado

La instrucción EXEC activará una acción local en su sistema local dependiendo de cómo haya configurado el generador de alarmas. Por ejemplo, se puede desactivar o activar una acción local. Si ha configurado el generador de alarmas de manera que envíe información a Operations Manager, las acciones locales normalmente no serán realizadas.

Ejemplos

En el ejemplo siguiente, la instrucción EXEC ejecuta el comando UNIX mailx cuando la métrica gbl_disk_util_peak excede el valor 20.

```
IF gbl_disk_util_peak > 20 THEN  
EXEC "echo 'high disk utilization detected'| mailx root"
```

El ejemplo siguiente muestra una instrucción EXEC que envía correo al administrador del sistema cuando la frecuencia de paquetes de red excede en promedio 1000 por segundo en 15 minutos.

```
ALARM gbl_net_packet_rate > 1000 for 15 minutes
  TYPE = "net busy"
  SEVERITY = 5
  START
  {
    RED ALERT "network is busy"
    EXEC "echo 'network busy condition detected' | mailx root"
  }
  END
  RESET ALERT "NETWORK OK"
```



Tenga cuidado al usar la instrucción EXEC con comandos o secuencias que tengan una alta sobrecarga si se ejecuta con frecuencia.

El generador de alarmas ejecuta el comando y espera hasta que finalice para continuar. Se recomienda que no se especifiquen comandos cuya conclusión requiera un largo periodo de tiempo.

Instrucción PRINT

La instrucción PRINT permite imprimir un mensaje en el programa `utility` con la función `analyze`. El generador de alarmas ignorará la instrucción PRINT.

Sintaxis:

PRINT *message*

- *message* — Una combinación de cadenas y valores numéricos usados para crear un mensaje. Los valores numéricos pueden ser formateados con los parámetros `[|[-]width[|decimals]]`. *Width* especifica el ancho del campo; *decimals* especifica el número de decimales que se va a usar. Los componentes alfanuméricos de un mensaje deberán estar entre comillas. Los valores numéricos están alineados a la derecha. El signo - (signo menos) especifica la alineación a la izquierda. Las cadenas alfanuméricas siempre están alineadas a la izquierda.

Ejemplo

```
PRINT "The total time the CPU was not idle is",
      gbl_cpu_total_time |6|2, "seconds"
```

Al ejecutarse, esta instrucción imprime un mensaje como el siguiente:

```
The total time the CPU was not idle is 95.00 seconds
```


Instrucción IF

Use la instrucción IF para definir una condición con la lógica IF-THEN (SI-ENTONCES) La instrucción IF debe ser usada con la instrucción ALARM. No obstante, también puede usarse sola o en cualquier lugar del archivo `alarmdef` en donde se necesite la lógica IF-THEN.

Si especifica una instrucción IF fuera de una instrucción ALARM, no tendrá control sobre la frecuencia con la que se ejecuta.

Sintaxis:

```
IF condition THEN action [ELSE action]
```

- **IF** *condition* —Una condición puede definirse como una comparación entre dos ítems.

```
item1 {>, <, >=, <=, ==, !=}item2  
[AND, OR[item3 {>, <, >=, <=, ==, !=}item4]]
```

en donde "==" significa "igual" y "!=" significa "no igual".

Un ítem puede ser un nombre de métrica, una constante numérica, una cadena alfanumérica entre comillas, un alias o una variable. Cuando se comparan cadenas alfanuméricas, sólo pueden utilizarse como operadores == o !=.

- *action* — Cualquier acción o establecer variable (ALARM no es válido en este caso).

Modo en que es utilizado

La instrucción IF comprueba la *condition*. Si la *condition* es true, la *action* después de THEN será ejecutada. Si la *condition* es false, entonces la *action* dependerá de la cláusula opcional ELSE. Si se ha especificado la cláusula ELSE, la *action* siguiente será ejecutada; de lo contrario, la instrucción IF no hará nada.

Ejemplo

En el ejemplo siguiente se calcula un síntoma de cuello de botella de CPU y se usa la probabilidad de cuello de botella resultante para definir alertas azul-verdoso o rojas. Dependiendo de cómo se haya configurado el generador de alarmas, la ALERT activará una captura SNMP en el Network Node Manager o el mensaje “End of CPU Bottleneck Alert” en Operations Manager junto con el porcentaje de CPU usado.

```
SYMPTOM CPU_Bottleneck > type=CPU  
  RULE gbl_cpu_total_util > 75 prob 25  
  RULE gbl_cpu_total_util > 85 prob 25  
  RULE gbl_cpu_total_util > 90 prob 25  
  RULE gbl_cpu_total_util > 4 prob 25  
ALARM CPU_Bottleneck > 50 for 5 minutes  
  TYPE="CPU"  
  START  
  IF CPU_Bottleneck > 90 then  
    RED ALERT "CPU Bottleneck probability= ",  
             CPU_Bottleneck, "%"  
  ELSE  
    CYAN ALERT "CPU Bottleneck probability= ",  
              CPU_Bottleneck, "%"  
  REPEAT every 10 minutes  
  IF CPU_Bottleneck > 90 then
```

```

        RED ALERT "CPU Bottleneck probability= ",
                CPU_Bottleneck, "%"
    ELSE
        CYAN ALERT "CPU Bottleneck probability= ",
                CPU_Bottleneck, "%"
    END
    RESET ALERT "End of CPU Bottleneck Alert"

```



No use métrica que esté registrada en intervalos diferentes en la misma instrucción. Por ejemplo, no itere un proceso (registrado en intervalos de 1 minuto) basado en el valor de una métrica global (registrada en intervalos de 5 minutos) en una instrucción como la que sigue:

```

IF global_metric THEN
    PROCESS LOOP ...

```

Los diferentes intervalos no serán sincronizados como se espera, de modo que los resultados no serán válidos.

Instrucción LOOP

La instrucción `LOOP` recorre los tipos de datos de instancias múltiples y ejecuta la *action* definida para cada instancia.

Sintaxis

```

{APPLICATION, PROCESS, LVOLUME, DISK, CPU, FILESYSTEM, TRANSACTION,
NETIF, LOGICAL}
LOOP
    action

```

- **APPLICATION, PROCESS, LVOLUME, DISK, CPU, FILESYSTEM, TRANSACTION, NETIF, LOGICAL** — tipos de datos de Componente Performance Collection que contienen datos multi-instancia.
- *action* — PRINT, EXEC, ALERT, establecer variables.

Modo en que es utilizado

A medida que las instrucciones `LOOP` iteran a través de cada instancia del tipo de datos, cambiarán los valores de métrica. Por ejemplo, la siguiente instrucción `LOOP` imprimirá el nombre de cada aplicación en `stdout` si usa el comando `analyze` del programa `utility`.

```

APPLICATION LOOP
    PRINT app_name

```

Un `LOOP` puede ser anidado en otra instrucción `LOOP` hasta un máximo de cinco niveles.

Para que se ejecute el `LOOP`, la instrucción `LOOP` deberá hacer referencia a una o más métricas del mismo tipo de datos definido en la instrucción `LOOP`.

Ejemplo

Use la instrucción `LOOP` para realizar un bucle por todas las aplicaciones activas.

El ejemplo siguiente muestra cómo determinar la aplicación que tiene el CPU más alto en cada intervalo.

```
highest_cpu = 0
APPLICATION loop
  IF app_cpu_total_util > highest_cpu THEN
  {
    highest_cpu = app_cpu_total_util
    big_app = app_name
  }
  ALERT "Application ", app_name, " has the highest cpu util at
",highest_cpu_util|5|2, "%"
  ALARM highest_cpu > 50
  START
    RED ALERT big_app, " is the highest CPU user at ", highest_cpu, "%"
  REPEAT EVERY 15 minutes
    CYAN ALERT big_app, " is the highest CPU user at ", highest_cpu, "%"
  END
  RESET ALERT "No applications using excessive cpu"
```

Instrucción INCLUDE

Use la instrucción `INCLUDE` para incluir otro archivo de definiciones de alarma junto con el archivo `alarmdef` predeterminado.

Sintaxis

```
INCLUDE "filename"
```

en donde *filename* es el nombre de otro archivo de definiciones de alarma. El nombre del archivo debe ser siempre un nombre completo.

Modo en que es utilizado

La instrucción `INCLUDE` puede ser usada para separar conjuntos de definiciones de alarma bien diferenciados lógicamente en archivos individuales.

Ejemplo

Por ejemplo, si tiene definiciones de alarma en un archivo individual para su métrica de transacción y éste se llama

```
trans_alarmdef1
```

podrá incluirlo agregando la línea siguiente a las definiciones de alarma de su archivo `alarmdef1`:

```
INCLUDE "/var/opt/perf/trans_alarmdef1"
```

Instrucción USE

Podrá agregar la instrucción `USE` para simplificar el uso de nombres de métrica en el archivo `alarmdef` cuando se referencien orígenes de datos distintos al origen de datos predeterminado `SCOPE`. Ello permite que se especifique un nombre de métrica sin tener que incluir el nombre del origen de datos.

El nombre del origen de datos deberá estar definido en el archivo `datasources`. El archivo `alarmdef` no podrá realizar la comprobación de sintaxis si detecta un nombre de origen de datos no disponible o inválido.

- ▶ El aspecto de una instrucción `USE` en el archivo `alarmdef` no implica que todos los nombres de métrica siguientes proceden del origen de datos especificado.

Sintaxis

```
USE "datasourcename"
```

Modo en que es utilizado

A medida que el generador de alarmas comprueba la validez de la sintaxis del archivo `alarmdef`, genera una lista de búsqueda por orden de todos los orígenes de datos que se referencian en el archivo. `Perfalarm` agrega secuencialmente entradas a esta lista de búsqueda de orígenes de datos a medida que detecta nombres de métrica completos o instrucciones `USE`. Esta lista es usada para hacer coincidir nombres de métrica no completos con el nombre de origen de datos correspondiente. La instrucción `USE` ofrece un cómodo método para agregar orígenes de datos a la lista de búsqueda de `perfalarm`, la cual a su vez permite acortar los nombres de métrica del archivo `alarmdef`. Para mayor información sobre la sintaxis de nombres de métrica consulte [Nombres de métrica](#) en la página 168 descrito con anterioridad en este capítulo.

El comportamiento predeterminado de `Perfalarm` para hacer coincidir nombres de métrica con un origen de datos es buscar primero el nombre de métrica en el origen de datos `SCOPE`. Esta instrucción implícita `USE "SCOPE"` es ejecutada cuando `perfalarm` detecta el primer nombre de métrica en el archivo `alarmdef`. Esta función habilita una ruta de búsqueda predeterminada al origen de datos `SCOPE` de manera que la métrica `SCOPE` pueda ser referenciada en el archivo `alarmdef` sin necesidad de tener un nombre completo.

A continuación se ofrece un ejemplo que ilustra lo anterior:

```
ALARM gbl_cpu_total_util > 80 FOR 10 MINUTES
    START RED ALERT "CPU utilization too high"
USE "ORACLE7"
ALARM ActiveTransactions >= 95 FOR 5 MINUTES
    START RED ALERT "Nearing limit of transactions for ORACLE7"
```

Cuando `perfalarm` comprueba la sintaxis del archivo `alarmdef` que contiene las instrucciones anteriores, detecta la métrica `"gbl_cpu_total_util"` y busca su origen de datos. `Perfalarm` no tiene aún ningún origen de datos en su lista de búsqueda de orígenes de datos, así que ejecuta una instrucción implícita `USE "SCOPE"` y a continuación busca el nombre de métrica en el origen de datos `SCOPE`. Encuentra la correspondencia y `perfalarm` continúa comprobando el resto del archivo `alarmdef`.

Cuando perfalarm detecta la instrucción USE "ORACLE7", agrega el origen de datos ORACLE7 a la lista de búsqueda de orígenes de datos. Cuando detecta el nombre de métrica "ActiveTransactions", perfalarm busca secuencialmente en la lista de orígenes de datos comenzando por el origen de datos SCOPE. SCOPE no contiene el nombre de métrica, así que prosigue con la búsqueda del origen de datos ORACLE7 y encuentra una correspondencia.

Si perfalarm no encuentra ninguna correspondencia con el nombre de métrica en el origen de datos, se imprimirá un mensaje de error y concluirá perfalarm.

Para cambiar el comportamiento de búsqueda predeterminado, puede agregarse una instrucción USE al principio del archivo alarmdef antes de las referencias a nombres de métrica. Ello hará que se agregue a la lista de búsqueda de orígenes de datos el origen de datos especificado en la instrucción USE antes del origen de datos SCOPE. Se realizará una búsqueda del origen(es) de datos en la instrucción(es) USE antes del origen de datos SCOPE para encontrar correspondencias con los nombres de métrica. A continuación se ilustra lo anterior con un ejemplo:

Una vez que se ha referenciado el origen de datos con una instrucción USE, no es posible cambiar el orden ni suprimirlo de la lista de búsqueda.

```
USE "ORACLE7"
ALARM gbl_cpu_total_util > 80 FOR 10 MINUTES
    START RED ALERT "CPU utilization too high"
ALARM ActiveTransactions >= 95 FOR 5 MINUTES
    START RED ALERT "Nearing limit of
transactions for ORACLE7"
```

En el ejemplo anterior ha cambiado el orden de las instrucciones en el archivo alarmdef. La instrucción USE "ORACLE7" ha sido definida antes de referenciarse nombres de métrica y, por lo tanto, el origen de datos ORACLE7 es agregado a la lista de búsqueda de orígenes de datos. La instrucción implícita USE "SCOPE" es ejecutada cuando perfalarm detecta el primer nombre de métrica en el archivo gbl_cpu_total_util. Dado que el nombre de métrica "gbl_cpu_total_util" no es un nombre completo, perfalarm buscará secuencialmente en la lista de orígenes de datos comenzando por ORACLE7. ORACLE7 no contiene el nombre de métrica, así que prosigue con la búsqueda del origen de datos SCOPE y encuentra una correspondencia.

Perfarm continúa comprobando el resto del archivo alarmdef. Cuando perfalarm detecta la métrica "ActiveTransactions", busca secuencialmente en la lista de orígenes de datos comenzando por ORACLE7. Encuentra la correspondencia y perfalarm continúa comprobando el resto del archivo alarmdef. Si perfalarm no encuentra una correspondencia con el nombre de métrica en ningún origen de datos, se imprimirá un mensaje de error y concluirá perfalarm.

Tenga precaución al usar la instrucción USE cuando múltiples orígenes de datos contengan nombres de métrica idénticos. Perfalarm busca secuencialmente en la lista de orígenes de datos. Si define condiciones de alarma de orígenes de datos distintos cuyos nombres de métrica sean idénticos, deberá completar los nombres de métrica con sus nombres de orígenes de datos correspondientes para garantizar que el valor de métrica sea capturado del origen de datos correcto. A continuación se ofrece un ejemplo en el que cada uno de los nombres de métrica de las instrucciones alarm incluye su origen de datos.

```
ALARM ORACLE7:ActiveTransactions >= 95 FOR 5 MINUTES
    START RED ALERT "Nearing limit of transactions for ORACLE7"
ALARM FINANCE:ActiveTransactions >= 95 FOR 5 MINUTES
    START RED ALERT "Nearing limit of transactions for FINANCE"
```

Instrucción VAR

La instrucción `VAR` permite definir una variable y asignarle un valor.

Sintaxis:

```
[VAR] name = value
```

- *name* — Las variables deben empezar por una letra e incluir letras, dígitos y el carácter de guión bajo. Los nombres de variables no distinguen entre mayúscula y minúscula.
- *value* — Si el valor es una cadena alfanumérica, deberá ir entre comillas.

Modo en que es utilizado

`VAR` asigna un valor a la variable del usuario. Si la variable no existe, se creará una.

Una vez definida, las variables pueden ser usadas en cualquier lugar del archivo `alarmdef`.

Ejemplos

Puede definir una variable asignándole algo a ella. La siguiente variable define la variable numérica *highest_CPU_value* asignándole un valor de cero.

```
highest_CPU_value = 0
```

El siguiente ejemplo define la variable alfanumérica *my_name* asignándole un valor de cadena nulo.

```
my_name = ""
```

Instrucción ALIAS

La instrucción `ALIAS` permite sustituir un alias si alguna parte de un nombre de métrica (clase, instancia o métrica) contiene un nombre que distingue entre mayúscula y minúscula o un nombre con caracteres especiales. Estas son las únicas circunstancias por las cuales usar la instrucción `ALIAS`.

Sintaxis

```
ALIAS name = "replaced-name"
```

- *name* — El nombre debe empezar por una letra e incluir letras, dígitos y el carácter de guión bajo.
- *replaced-name* — Nombre que será reemplazado por la instrucción `ALIAS` para que sea unívocamente reconocible por el generador de alarmas.

Modo en que es utilizado

Debido al modo en que es procesado el archivo `alarmdef`, si alguna parte de un nombre de métrica (clase, instancia o nombre de métrica) sólo puede ser identificada unívocamente distinguiendo entre mayúsculas y minúsculas, deberá crearse un alias. Deberá también crearse un alias en el caso de nombres con caracteres especiales. Por ejemplo, si tiene aplicaciones llamadas "BIG" y "big," deberá crear un alias "big" para asegurarse de que son identificadas como aplicaciones diferentes. El alias deberá ser definido en alguna parte del archivo `alarmdef` con anterioridad a la *primera* instancia del nombre que se desea sustituir.

Ejemplos

Dado que no es posible utilizar caracteres especiales ni mayúsculas o minúsculas en la sintaxis, produciría errores usar el nombre de aplicación "AppA" y "appa" porque el procesamiento no distinguiría uno del otro. Debería en ese caso crear un alias "AppA" para asignarle un nombre reconocible unívoco. Por ejemplo:

```
ALIAS appa_uc = "AppA"  
ALERT "CPU alert for AppA.util is", appa_uc:app_cpu_total_util
```

Si usa un alias para una instancia con un identificador de clase, incluya en el alias tanto el nombre de instancia como el de clase. El siguiente ejemplo muestra el alias del nombre de instancia 'other' y el nombre de clase 'APPLICATION'.

```
ALIAS my_app="other (APPLICATION) "  
ALERT my_app:app_cpu_total_util > 50 for 5 minutes
```

Instrucción SYMPTOM

Un `SYMPTOM` es un modo de establecer un valor único variable en función de un conjunto de condiciones. Cuando una de las condiciones es true, su valor de probabilidad es agregado al valor de la variable `SYMPTOM`.

Sintaxis

```
SYMPTOM variable  
RULE condition PROB probability  
[RULE condition PROB probability]  
.  
.  
.
```

- Las palabras clave **SYMPTOM** y **RULE** son usadas de manera exclusiva en la instrucción `SYMPTOM` y no pueden ser usadas en otras instrucciones de sintaxis. La instrucción `SYMPTOM` debe ser una instrucción de nivel superior y no puede ser anidada en otra instrucción. Ninguna otra instrucción podrá seguir a `SYMPTOM` hasta que todas sus instrucciones `RULE` correspondientes hayan concluido.
- *variable* es un nombre de variable que será el nombre de este síntoma. Los nombres de variables definidos en la instrucción `SYMPTOM` pueden ser usados en otras instrucciones de sintaxis, pero el valor de variable no deberá ser cambiado en esas instrucciones.
- **RULE** es una opción de la instrucción `SYMPTOM` y no puede ser usada independientemente. En la instrucción `SYMPTOM` se pueden usar tantas opciones **RULE** como se necesiten. La variable `SYMPTOM` es evaluada de acuerdo a las reglas en cada intervalo.
- Una *condition* puede definirse como una comparación entre dos ítems.

```
item1 {>, <, >=, <=, ==, !=}item2  
[item3 {>, <, >=, <=, ==, !=}item4]
```

en donde "==" significa "igual" y "!=" significa "no igual".

Un ítem puede ser un nombre de métrica, una constante numérica, una cadena alfanumérica entre comillas, un alias o una variable. Cuando se comparan ítems alfanuméricos, sólo pueden utilizarse como operadores == o !=.

- *probability* es una constante numérica. Las probabilidades de todas las SYMPTOM RULE que son true son agregadas conjuntamente para crear un valor SYMPTOM.

Modo en que es utilizado

La suma de todas las probabilidades en las que la condición entre medida (measurement) y valor (value) es true es la probabilidad de que ocurra el síntoma.

Ejemplo

```
SYMPTOM CPU_Bottleneck  
RULE gbl_cpu_total_util > 75   PROB 25  
RULE gbl_cpu_total_util > 85   PROB 25  
RULE gbl_cpu_total_util > 90   PROB 25  
RULE gbl_run_queue > 3        PROB 50  
IF CPU bottleneck > 50 THEN  
CYAN ALERT "The CPU symptom is: ", CPU_bottleneck
```

Ejemplos de definiciones de alarma

Los siguientes ejemplos muestran los usos más comunes de definiciones de alarma.

Ejemplo de un problema de CPU

Dependiendo del modo en que haya configurado el generador de alarmas, este ejemplo activará una captura SNMP a Network Node Manager o un mensaje a Operations Manager cuando el uso de CPU supere el valor de 90% durante 5 minutos y la cola de ejecución de CPU supere el valor 3 durante 5 minutos.

```
ALARM gbl_cpu_total_util > 90 AND  
      gbl_run_queue > 3 FOR 5 MINUTES  
START  
      CYAN ALERT "CPU too high at", gbl_cpu_total_util, "%"  
REPEAT EVERY 20 MINUTES  
{  
      RED ALERT "CPU still to high at ", gbl_cpu_total_util, "%"  
      EXEC "/usr/bin/pager -n 555-3456"  
}  
END  
      RESET ALERT "CPU at ", gbl_cpu_total_util, "% - RELAX"
```

Si ambas condiciones continúan siendo true durante 20 minutos, se podrá generar una alarma de gravedad crítica en la ventana del explorador de alarmas (Alarm Browser) de Network Node Manager. A continuación se ejecutará un programa para establecer contacto con el administrador del sistema.

Cuando alguna de las condiciones de alarma deje de ser true, el símbolo de alarma será eliminado y se enviará un mensaje que muestra el uso de CPU, la hora en la que concluyó la alarma y una nota a RELAX.

Ejemplo de uso de Swap

En el siguiente ejemplo ALERT activará el envío de una captura SNMP a Network Node Manager o el envío de un mensaje a Operations Manager dependiendo del modo en que haya configurado el generador de alarmas siempre que el uso del espacio swap exceda el valor de 95% durante 5 minutos.

```
ALARM gbl_swap_space_util > 95 FOR 5 MINUTES
  START
    RED ALERT "GLOBAL SWAP space is nearly full "
  END
  RESET ALERT "End of GLOBAL SWAP full condition"
```

Ejemplo de alarmas basadas en horas

Es posible especificar el intervalo de tiempo en el que las condiciones de alarma permanecerán activas. Por ejemplo, si está ejecutando trabajos de mantenimiento del sistema programados a intervalos regulares, podrá especificar las condiciones de alarma de horas de funcionamiento habituales y un conjunto diferente de condiciones de alarma exclusivo para el periodo (horas) de mantenimiento del sistema.

En el siguiente ejemplo la alarma sólo será activada durante el día, entre las 8:00AM y las 5:00PM.

```
start_shift = "08:00"
end_shift = "17:00"
ALARM gbl_cpu_total_util > 80
  TIME > start_shift
  TIME < end_shift for 10 minutes
  TYPE = "cpu"
  START
    CYAN ALERT "cpu too high at ", gbl_cpu_total_util, "%"
  REPEAT EVERY 10 minutes
    RED ALERT "cpu still too high at ", gbl_cpu_total_util, "%"
  END
  IF time == end_shift then
  {
  IF gbl_cpu_total_util > 80 then
    RESET ALERT "cpu still too high, but at the end of shift"
  ELSE
    RESET ALERT "cpu back to normal"
  ELSE
```

Ejemplo de alarmas de instancias de disco

Las alarmas pueden ser generadas para un disco en particular identificando el nombre de instancia del disco y el nombre de métrica correspondiente.

El siguiente ejemplo de sintaxis de alarma genera alarmas para una instancia de disco determinada. Es preciso crear alias si la instancia de disco contiene caracteres especiales.

```
ALIAS diskname=""
ALARM diskname:bydisk_phys_read > 1000 for 5 minutes
TYPE="Disk"
START
  RED ALERT "Disk  "
REPEAT EVERY 10 MINUTES
  CYAN ALERT "Disk  cyan alert"
END
  RESET ALERT "Disk  reset alert"
```

Personalización de definiciones de alarma

Las condiciones para generar alarmas pueden ser especificadas por el usuario en el archivo `alarmdef`. Cuando se instale por primera vez Componente Performance Collection, el archivo `alarmdef` contendrá un conjunto de definiciones de alarma predeterminadas. Puede usar estas definiciones de alarma predeterminadas o personalizarlas según le convenga.

Para personalizar el archivo `alarmdef` realice lo sigue:

- 1 Compruebe su definición de alarma(s) según se requiera. Consulte los diversos ejemplos de sintaxis de definiciones de alarma si lo cree necesario.
- 2 Guarde el archivo.
- 3 Valide las definiciones de alarma con ayuda del programa `utility` de Componente Performance Collection:
 - a Escriba **utility**.
 - b Cuando se muestre el símbolo del sistema, escriba:
checkdef

Con ello se comprobará la sintaxis de alarma y se mostrarán errores o advertencias si hubiese algún problema con el archivo.
- 4 Para que las nuevas definiciones de alarma surtan efecto, escriba:
ovpa restart alarm
o
mwa restart alarm

Con ello se detendrá y reiniciará el generador de alarmas, y se leerá el archivo personalizado `alarmdef`.

Podrá usar un conjunto exclusivo de definiciones de alarmas para cada sistema de Componente Performance Collection u optar por estandarizar la monitorización de un grupo de sistemas usando el mismo conjunto de definiciones de alarmas para dicho grupo.

Si el archivo `alarmdef` es muy extenso, puede que el generador de alarmas de Componente Performance Collection y los programas `utility` no funcionen según lo esperado. Este problema ocurrirá o no en función de la disponibilidad de recursos del sistema.

El mejor modo de adquirir experiencia sobre las alarmas de rendimiento es practicar agregando definiciones de alarma o cambiando las definiciones de alarma predeterminadas.

10 Asesor del componente RTMA

La función de asesor sólo puede usarse si se habilita HP Ops OS Inst to Realtime Inst LTU o Glance Pak Software LTU.

La función de asesor permite al usuario generar y ver alarmas cuando los valores de determinadas métricas, recopiladas por el componente RTMA, exceden el valor de umbral establecido (o se encuentran por debajo del mismo). La **secuencia de comandos de asesor** y la utilidad **padv** conforman la función de asesor. La secuencia de asesor permite crear las reglas para generar alarmas cuando el rendimiento del sistema monitorizado muestre signos de degradación. La utilidad `padv` permite ejecutar la secuencia de asesor en el sistema deseado.



El tema siguiente se centra en el uso de la función de asesor con el componente RTMA. El software GlancePlus ofrece funciones adicionales que pueden ser usadas con la utilidad de asesor. Para información sobre el uso de la función de asesor con el software GlancePlus, consulte *online help for GlancePlus*.

Alarmas y síntomas

Las alarmas permiten resaltar condiciones de métrica. La secuencia de asesor permite definir los valores de umbral de métricas que son monitorizadas por el componente RTMA. Cuando el valor de métrica excede el valor de umbral preestablecido, el componente RTMA genera una alarma en forma de mensaje de alerta. Este mensaje es enviado en forma de `stdout` a la utilidad **padv**.

Una alarma puede ser activada cuando se cumplen las condiciones especificadas por el usuario. Las alarmas se basan en periodos de tiempo especificados por el usuario que pueden ser de un intervalo o más largos.

Un síntoma es una combinación de condiciones que afecta el rendimiento del sistema.

Observando diversas métricas con sus valores de umbral correspondientes y agregando valores a la probabilidad de que estas métricas contribuyan a un cuello de botella, el asesor calcula un valor que representa la probabilidad combinada de que exista un cuello de botella.

Funcionamiento de la secuencias de comandos de asesor

Al ejecutar el comando `padv`, HP Operations Agent digitaliza la secuencia especificada con el comando y actúa consecuentemente. Si el usuario no especifica ningún archivo de secuencia de comando con el comando `padv`, la utilidad de asesor recuperará la información necesaria en el archivo de secuencia predeterminado:

- *En Windows:* `%ovdatadir%\perf\perfd`
- *En UNIX y Linux:* `/var/opt/perf/perfd`

Para ejecutar una secuencia de comandos que incluya diagnóstico y acciones específicas del sistema operativo, deberán usarse las secuencias predeterminadas siguientes:

- *En Windows:* `%ovdatadir%\perf\perfd\os\<os_type>\adv`
- *En UNIX y Linux:* `/var/opt/perf/perfd/os/<os_type>/adv`

En este caso, `<os_type>` es el tipo de sistema operativo de nodo en el que se desea ejecutar la secuencia.

Tras ejecutar la secuencia de asesor se pueden realizar las siguientes acciones:

- Imprimir el estado del sistema en función de las alarmas generadas en un archivo de texto.
- Ver el estado en tiempo real del sistema en la consola de comandos en la que se ha ejecutado el comando `padv`.

Uso del asesor

Para usar el componente de asesor con objeto de monitorizar el estado del sistema en tiempo real, realice lo siguiente:

- 1 Configure la secuencia de asesor según sus requisitos. El directorio siguiente contiene secuencias de ejemplo:
 - *En Windows:* `%ovinstalldir%\examples\adviser`
 - *En UNIX o Linux:* `/opt/perf/examples/adviser`
- 2 Identifique el nodo en el que se desea ejecutar la secuencia.
- 3 Asegúrese de que el proceso `perfd` está siendo ejecutado en el sistema identificado.
- 4 Ejecute el comando siguiente:

```
padv -s <script_name> -n <system_name>
```

La secuencia de asesor comienza a ejecutarse en el sistema especificado y genera resultados basados en la configuración del archivo de secuencia.

Ejecución de la secuencia de asesor en múltiples sistemas

El comando `mpadv` permite ejecutar la secuencia de asesor en múltiples sistemas. Para usar el comando `mpadv`, siga los pasos siguientes:

- 1 Identifique los nodos en los que se desea ejecutar la secuencia.
- 2 Cree un archivo de texto que contenga un listado de los nombres de los sistemas identificados.
- 3 Guarde el archivo de texto en el sistema local.
- 4 Configure la secuencia de asesor según sus requisitos. El directorio siguiente contiene secuencias de ejemplo:

- *En Windows:* `%ovinstalldir%\examples\adviser`
- *En UNIX o Linux:* `/opt/perf/examples/adviser`

- 5 Asegúrese de que el proceso `perfd` está siendo ejecutado en el sistema identificado.
- 6 Ejecute el comando siguiente:

```
mpadv -l <system_list_text_file> -s <script_name>
```

La secuencia de asesor comienza a ejecutarse en los sistemas especificados en el archivo `<system_list_text_file>` y genera resultados basados en la configuración del archivo de secuencia.

Sintaxis del asesor

La sintaxis del asesor es un sencillo lenguaje de secuencias de comandos que permite al usuario establecer alarmas y definir condiciones de síntomas.

El siguiente directorio contiene un archivo de sintaxis predeterminado— `adviser.syntax`—:

- *En Windows:* `%ovdatadir%\perf`
- *En UNIX y Linux:* `/var/opt/perf`

El archivo de sintaxis puede ser editado para especificar alarmas y síntomas del usuario.

Convenciones de sintaxis

- Las llaves de apertura y cierre (`{ }`) indican que una de las opciones es requerida.
- Los corchetes (`[]`) indican un ítem opcional.
- Ítems separados por comas con corchetes o llaves son opciones. Seleccione sólo una.
- Las cursivas indican un nombre de variable que el usuario reemplazará.
- Las palabras clave de sintaxis del asesor deben escribirse siempre en mayúscula.

Comentarios

Sintaxis:

```
# [cualquier texto o caracteres]
```

o bien

```
// [cualquier texto o caracteres]
```

Los comentarios pueden estar precedidos por bien las barras diagonales (//) o el signo de almohadilla (#). En ambos casos el comentario terminará al final de la línea.

Condiciones

Una condición se puede definir como una comparación entre dos nombres de métrica, variables del usuario o constantes numéricas.

```
item1 {>, <, >=, <=, ==, !=} item2 [OR item3 \  
{>, <, >=, <=, ==, !=} item4]
```

o bien

```
item1 {>, <, >=, <=, ==, !=} item2 [AND item3 \  
{>, <, >=, <=, ==, !=} item4]
```

("==" significa "igual", y "!=" significa "no igual".)

Las condiciones son utilizadas en la instrucción ALARM y en la instrucción IF. Permiten comparar dos métricas numéricas, variables o constantes, y pueden usarse también entre dos nombres de métrica de cadena, variables del usuario o constantes de cadena. Para condiciones de cadena, sólo pueden usarse como operadores == or !=.

Se pueden usar condiciones compuestas especificando los operadores OR o AND entre subcondiciones.

Ejemplos:

```
gbl_swap_space_reserved_util > 95
```

```
proc_proc_name == "test" OR proc_user_name == "tester"
```

```
proc_proc_name != "test" AND
```

```
proc_cpu_sys_mode_util > highest_proc_so_far
```


Constantes

Las constantes pueden ser tanto alfanuméricas como numéricas. Una constante alfanumérica debe estar entre comillas dobles. Existen dos tipos de constantes numéricas: enteras y reales. Las constantes enteras pueden contener sólo dígitos y un indicador de signo opcional. Las constantes reales pueden incluir también un punto decimal.

Ejemplos:

345	Entero numérico
345.2	Real numérico
“Time is”	Literal alfanumérico

Expresiones

Use expresiones para evaluar valores numéricos. Una expresión puede ser usada en una condición o en una acción.

Una expresión puede contener:

- Constantes numéricas
- Nombres de métrica numéricos
- Variables numéricas
- Una combinación aritmética de lo anterior
- Una combinación de lo anterior entre paréntesis.

Ejemplos:

```
Iteration + 1
```

```
3.1416
```

```
gbl_cpu_total_util - gbl_cpu_user_mode_util  
( 100 - gbl_cpu_total_util ) / 100.0
```

Nombres de métrica en sintaxis del asesor

Se pueden referenciar directamente métricas en cualquier lugar de la sintaxis del asesor. Se pueden usar los siguientes tipos de métrica en la sintaxis del asesor:

- Métricas globales (precedidas por el prefijo gbl_ o tbl_)
- Métricas de aplicación (precedidas por el prefijo app_)
- Métricas de proceso (precedidas por el prefijo proc_)
- Métricas de disco (precedidas por el prefijo bydisk_)
- Métricas por CPU (precedidas por el prefijo bycpu_)
- Métricas de sistema de archivos (precedidas por el prefijo fs_)
- Métricas de volumen lógico (precedidas por el prefijo lv_)
- Métricas de interfaz de red (precedidas por el prefijo bynetif_)

- Métricas de intercambio (precedidas por el prefijo `byswp_`)
- Métricas ARM (precedidas por el prefijo `tt_` o `ttbin_`)
- Métricas PRM (precedidas por el prefijo `prm_`)
- Métricas de dominio de localidad (precedidas por el prefijo `ldom_`)

Sólo pueden utilizarse métricas de proceso, volumen lógico, disco, sistema de archivos, LAN e intercambio dentro del contexto de una instrucción LOOP.

Las métricas pueden contener datos alfanuméricos (por ejemplo, `gbl_machine` o `app_name`) o datos numéricos y pueden reflejar diferentes tipos de medidas. Por ejemplo, la métrica que termina en un nombre de métrica indica lo que está siendo medido:

- la métrica a `_util` mide la utilización en porcentajes
- la métrica a `_rate` mide las unidades por segundo
- la métrica a `_queue` mide el número de procesos o subprocesos que esperan un recurso

En caso de no tener certeza sobre la unidad de medida correspondiente a una métrica determinada, consulte el documento de definiciones de métrica.

Deberá asociarse una métrica de aplicación a una aplicación determinada, excepto cuando se use la instrucción LOOP. Para ello, especifíquese el nombre de aplicación seguido de una coma, y a continuación el nombre de métrica. Por ejemplo, `other_apps:app_cpu_total_util` especifica la utilización total de CPU de la aplicación `other_apps`. Consulte la descripción de la instrucción ALIAS para mayor información sobre el uso de métricas de aplicación en la sintaxis.

Los nombres de aplicación, tal y como se definen en el archivo `parm`, pueden contener caracteres especiales y espacios en blanco insertados. Para usar estos nombres en la sintaxis (en donde los nombres de aplicación deberán corresponderse con la forma de un nombre de variable), en los nombres no se establece distinción entre mayúsculas y minúsculas y los espacios en blanco insertados son convertidos en guiones bajos. Ello significa que el nombre de la aplicación definido como "Other Apps" puede ser referenciado en la sintaxis como `other_apps`. En el caso de nombres de aplicación definidos con caracteres especiales, deberá usarse la instrucción ALIAS para especificar un nombre alternativo.

En el caso de métricas de aplicación explícitamente calificadas, éstas pueden ser referenciadas en cualquier lugar de la sintaxis. En el caso de métricas de aplicación no calificadas, éstas sólo podrán ser referenciadas dentro del contexto de la instrucción LOOP. Ésta es una instrucción iterativa que califica implícitamente métricas de proceso o aplicación.

Sólo podrán referenciarse métricas de proceso dentro del contexto de una instrucción LOOP. No es posible referenciar explícitamente un proceso.

Printlist

`printlist` es una combinación de nombres de métrica, variables del usuario, constantes o expresiones adecuadamente formateadas. Consulte los ejemplos para obtener el formato adecuado.

Ejemplos de expresiones:

```
expression [|width[|decimals|]]
```

Metric Names or User Variable examples:

```
metric names [|width[|decimals|]]
```

o bien

user variables [|width[|decimals]]

Los nombres de métrica o variables del usuario deben ser alfanuméricos.

Ejemplos de constantes:

No formatting is necessary for constants.

Ejemplos de formato:

```
gbl_cpu_total_util|6|2    formats as  '100.00'  
(100.32 + 20) |6    formats as      ' 120'  
gbl_machine|5    formats as          '7013/'  
"User Label"      formats as          "User Label"
```

Variables

Las variables deben empezar por una letra e incluir letras, dígitos y el caracter de guión bajo. Las variables no distinguen entre mayúscula y minúscula.

Defina una variable asignándole algo a ella. La siguiente variable define la variable numérica `highest_CPU_value` asignándole un valor de cero.

```
highest_CPU_value = 0
```

El siguiente ejemplo define la variable alfanumérica `my_name` asignándole un valor de cadena nulo.

```
my_name = ""
```

Instrucciones de sintaxis de asesor

Instrucción ALARM

El usuario podrá usar la instrucción `ALARM` para ser notificado cuando ciertos eventos, definidos por él mismo, ocurran en el sistema. Por medio de la instrucción `ALARM`, la secuencia de asesor puede notificar al usuario por medio de mensajes enviados a la consola de origen del comando `padv`.

Sintaxis:

```
ALARM condition [FOR duration {SECONDS, MINUTES, INTERVALS}]  
    [condition [FOR duration {SECONDS, MINUTES, INTERVALS}] ] ...
```

```
[START statement]
```

```
[REPEAT [EVERY duration [SECONDS, MINUTES, INTERVAL, INTERVALS]]  
    statement]
```

```
[END statement]
```

La instrucción **ALARM** debe ser una instrucción de nivel superior. No puede ser anidada en ninguna otra instrucción.

No obstante, el usuario podrá incluir varias condiciones **ALARM** en una misma instrucción **ALARM**, en cuyo caso todas las condiciones deberán establecerse en true para que la alarma se active. El usuario podrá además usar una instrucción **COMPOUND**, la cual será ejecutada en su debido momento durante el ciclo de alarma.

START, **REPEAT** y **END** son palabras clave de la instrucción **ALARM**. Cada una de estas palabras clave especifica una instrucción. El usuario deberá tener la palabra clave **START**, **REPEAT** o **END** en una instrucción **ALARM**, y dichas palabras claves deberán aparecer en el orden correcto.

El ciclo de alarma comienza en el primer intervalo en el que todas las condiciones de alarma son true durante al menos la duración especificada. En dicho momento, la secuencia de asesor ejecuta una instrucción **START**, y en cada intervalo siguiente comprueba la condición **REPEAT**. Si ha transcurrido el tiempo suficiente, se ejecuta la instrucción de la cláusula **REPEAT**. Este proceso continua hasta que una o más condiciones de alarma sea false. Con ello finaliza el ciclo de alarma y es ejecutada la instrucción **END**.

Si se omite la especificación **EVERY** en la instrucción **REPEAT**, la secuencia de asesor ejecutará la instrucción **REPEAT** en cada intervalo.

Ejemplo de **ALARM**: Instrucción **ALARM** típica

El siguiente ejemplo de **ALARM** establece una alerta roja cuando la tabla de semáforo esté casi llena. Es similar a una alarma predefinida en la sintaxis predeterminada. No lo agregue a su sintaxis sin haber eliminado antes el predeterminado, o sus subsiguientes mensajes de alerta podrán ser confusos.

```
ALARM tbl_sem_table_util > 90 FOR 1 MINUTE

    START RED ALERT "Semaphore Table is nearly full"

    REPEAT EVERY 30 SECONDS

        RED ALERT "Semaphore Table still nearly full"

    END RESET ALERT "End of Semaphore Table full condition"
```

Este ejemplo de **ALARM** prueba la métrica `tbl_sem_table_util` para determinar si es superior a 90. Si lo es, la instrucción **RED ALERT** enviará un mensaje con nivel de gravedad **RED** a la consola de origen del comando `padv`.

La instrucción **REPEAT** comprueba la condición `tbl_sem_table_util` cada 30 segundos. Siempre y cuando la condición sea superior a 90, **REPEAT** instruirá al asesor que mantenga la condición de **RED ALERT** y envíe el mensaje `Semaphore Table still nearly full` a la consola de origen del comando `padv`.

Cuando la condición `tbl_sem_table_util` sea inferior a 90, la instrucción **RESET ALERT** mostrará el mensaje `End of Semaphore Table full condition` en la consola de origen del comando `padv`.

Ejemplo de ALARM: Uso de condiciones múltiples

Es posible tener más de una condición de prueba en la instrucción ALARM. En dicho caso, todas las instrucciones deberán ser true para que el botón de alarma se active. Por ejemplo:

```
ALARM gbl_cpu_total_util > 90 FOR 2 MINUTES
    gbl_cpu_sys_mode_util > 50 FOR 1 MINUTES
    START RED ALERT
        "The CPU is busy and System Mode CPU utilization is high."
    END RESET ALERT "The CPU alert is now over."
```

Este ejemplo de ALARM prueba la métrica `gbl_cpu_total_util` y `CPU_Bottleneck`. Si ambas condiciones son true, la instrucción RED ALERT establece una alerta crítica. Si una de las condiciones de prueba es false, la instrucción RESET es ejecutada.

Ejemplo de ALARM : espacio de intercambio (swap)

```
//GLOBAL SWAP ALARM
symp_swap_util = gbl_swap_space_used / gbl_swap_space_avail
ALARM symp_swap_util > 0.9
    START
        RED ALERT "GLOBAL SWAP space is nearly full"
    END RESET ALERT "GLOBAL SWAP space crisis is over"
```

La nueva variable, `symp_swap_util`, representa la utilización de espacio swap. La secuencia de asesor crea una alarma cuando la utilización de espacio swap es superior al 90%. En el siguiente intervalo, si `symp_swap_util` cae por debajo del 90%, la condición de alarma cambia a false, y ALARM es reestablecida.

Ejemplo de ALARM: Alerta amarilla

```
ALARM Symp_Global_Cpu_Bottleneck > 50 FOR 2 MINUTES

    START YELLOW ALERT "CPU Bottleneck probability= ",
        Symp_Global_Cpu_Bottleneck, "% for the last 2 minutes"

    REPEAT every 2 minutes
        YELLOW ALERT "CPU Bottleneck probability= ",
            Symp_Global_Cpu_Bottleneck, "% for the last 2 minutes"

    END

    RESET ALERT " CPU Bottleneck Yellow Alert over, probability=",
        Symp_Global_Cpu_Bottleneck, "%"
```

ALARM prueba la variable SYMPTOM, la cual es definida en la instrucción SYMPTOM `Symp_Global_Cpu_Bottleneck`. Si la variable SYMPTOM es superior a 50 durante dos minutos, ALARM notificará al usuario con una YELLOW ALERT en la consola de comandos `padv`.

ALARM se repetirá cada 2 minutos hasta que la condición ALARM se establezca en false. En dicho momento, la instrucción END reestablece ALERT.

Ejemplo de ALARM: Problema de CPU

Ejemplo de

```
gbl_cpu_total_util > 90 FOR 30 SECONDS

gbl_run_queue > 3 FOR 30 SECONDS

START YELLOW ALERT "CPU AT ", gbl_cpu_total_util,
"% at ", gbl_stattime

REPEAT EVERY 300 SECONDS {
  RED ALERT "CPU AT ", gbl_cpu_total_util
  exec "/usr/bin/pager -n 555-3456"
}

END ALERT "CPU at ", gbl_cpu_total_util, "% at ",
gbl_stattime, " - RELAX"
```

El ejemplo anterior genera una alerta amarilla en la consola de comandos `padv` cuando la utilización del CPU sea superior al 90% durante 30 segundos y la cola de ejecución del CPU sea superior a 3 durante 30 segundos.

Si ambas condiciones continúan true, se generará una alerta roja y se invocará un programa para llamar al administrador del sistema.

Instrucción ALERT

La instrucción ALERT permite poner un mensaje en la consola de comandos `padv`. Cuando ALARM detecte un problema, ejecutará una instrucción ALERT para enviar un mensaje con el nivel de gravedad especificado a la consola de comandos `padv`.

La instrucción ALERT puede utilizarse junto con la instrucción ALARM.

Sintaxis:

```
[(RED or CRITICAL), (YELLOW or WARNING), RESET] ALERT printlist
```

RED y YELLOW son sinónimos de CRITICAL y WARNING.

Ejemplo de ALERT

Exemplo de instrucción ALERT:

```
RED ALERT "CPU utilization = ", gbl_cpu_total_util,
" at ", gbl_stattime
```

Al ejecutar esta instrucción se escribirá un mensaje en la consola de comandos `padv` que muestra, por ejemplo, lo siguiente:

```
CPU utilization = 85.6 at 14:43:10
```

Instrucción ALIAS

Use la instrucción `ALIAS` para asignar una variable a un nombre de aplicación que contenga caracteres especiales o espacios en blanco insertados.

Sintaxis:

```
ALIAS variable = "alias name"
```

```
ALIAS Example
```

Dado que no es posible usar caracteres especiales ni espacios en blanco insertados en la sintaxis, provocaría un error usar el nombre de aplicación "other user root" en la instrucción `PRINT` que aparece a continuación. Con `ALIAS`, el usuario puede usar en la sintaxis "other user root" y otras cadenas con caracteres especiales y espacios en blanco.

```
ALIAS otherapp = "other user root"
```

```
PRINT "CPU for other root login processes is:",  
      otherapp:app_cpu_total_util
```

Instrucción ASSIGNMENT

Use la instrucción `ASSIGNMENT` para asignar una expresión o valor numérico o alfanumérico a la variable del usuario.

Sintaxis:

```
[VAR] variable = expression
```

```
[VAR] variable = alphaitem
```

```
[VAR] variable = alphaitem
```

Ejemplos de ASSIGNMENT

En la primera instrucción se determina si la variable del usuario es numérica o alfanumérica. No es posible mezclar variables de tipos diferentes en una instrucción de asignación.

El siguiente ejemplo asigna un nombre de aplicación alfanumérico a una nueva variable de usuario:

```
myapp_name = other:app_name
```

El siguiente ejemplo es incorrecto ya que se asigna un valor numérico a una variable de usuario que anteriormente ha sido definida como alfanumérica (en el ejemplo 1):

```
myapp_name = 14
```

El siguiente ejemplo asigna un valor numérico a una nueva variable de usuario:

```
highest_cpu = gbl_cpu_total_util
```

El siguiente ejemplo es incorrecto ya que se asigna un valor alfanumérico literal a una variable de usuario que anteriormente ha sido definida como numérica (en el ejemplo 3):

```
highest_cpu = "Time is"
```

Instrucción COMPOUND

Use la instrucción COMPOUND con las instrucciones IF y LOOP, y las cláusulas START, REPEAT y END de la instrucción ALARM. Al usar una instrucción COMPOUND es ejecutada una lista de instrucciones.

Sintaxis

```
{  
statement  
statement  
}
```

Cree instrucciones compound agrupando una lista de instrucciones entre llaves ({}). La instrucción compound puede ser tratada como una instrucción simple en la sintaxis.

Las instrucciones compound no pueden incluir instrucciones ALARM y SYMPTOM. (Compound es un tipo de instrucción y no una palabra clave).

Ejemplo de COMPOUND

```
highest_cpu = highest_cpu  
IF gbl_cpu_total_util > highest_cpu THEN  
  // Begin compound statement  
  {  
    highest_cpu = gbl_cpu_total_util  
    PRINT "Our new high CPU value is ", highest_cpu, "%"  
  }  
  // End compound statement
```

En el ejemplo anterior, `highest_cpu = highest_cpu` define una variable llamada `highest_cpu`. La secuencia de comandos de asesor guarda el valor `highest_cpu` y notifica al usuario sólo cuando ese valor `highest_cpu` sea superado por un valor más alto.

En el ejemplo anterior, si se ha sustituido `highest_cpu = highest_cpu` por `highest_cpu = 0`, entonces el valor `highest_cpu` será reestablecido a cero en cada intervalo.

El usuario será notificado del valor de `highest_cpu` en cada intervalo.

Instrucción EXEC

Use la instrucción EXEC para ejecutar un comando UNIX desde su sintaxis de asesor. Podrá usarse el comando EXEC si, por ejemplo, el usuario desea enviar un mensaje por email al personal de MIS cada vez que se cumpla determinada condición.

Sintaxis

```
EXEC printlist
```

La `printlist` resultante será enviada al sistema operativo para su ejecución.

Dado que el comando EXEC especificado por el usuario puede ejecutarse una vez por intervalo de actualización, debe tenerse cuidado al usar este comando con secuencias o comandos de sistemas operativos que tengan una alta sobrecarga.

Ejemplos de EXEC

En el ejemplo siguiente EXEC ejecuta el comando UNIX mailx en cada intervalo.

```
EXEC "echo 'gpm mailed you a message' | mailx root"
```

En el ejemplo siguiente, EXEC sólo ejecuta el comando UNIX mailx cuando la métrica gbl_disk_util_peak supere el valor de 20.

```
IF gbl_disk_util_peak > 20 THEN
    EXEC "echo 'gpm detects high disk utilization' | mailx root"
```

Instrucción IF

Use la instrucción IF para probar condiciones definidas por el usuario en la sintaxis de comandos de asesor.

Sintaxis:

```
IF condition THEN statement [ELSE statement]
```

La instrucción IF prueba la condición. Si la condición es true, la instrucción siguiente a THEN será ejecutada. Si la condición es false, entonces la acción dependerá de la cláusula opcional ELSE.

Si se ha especificado una cláusula ELSE, se ejecutará la instrucción siguiente. De lo contrario, la instrucción IF no hará nada. La instrucción puede ser una instrucción COMPOUND que indica al comando de asesor que ejecute instrucciones múltiples.

Ejemplo de IF

```
IF gbl_cpu_total_util > 90 THEN
    PRINT "The CPU is running hot at: ", gbl_cpu_total_util
ELSE IF gbl_cpu_total_util < 20 THEN
    PRINT "The CPU is idling at: ", gbl_cpu_total_util
```

En este ejemplo la instrucción IF comprueba la condición (gbl_cpu_total_util > 90). Si la condición es true, entonces "The CPU is running hot at: " se mostrará en la consola de comandos padv junto con el % de CPU usado.

Si la condición (gbl_cpu_total_util > 90) es false, ELSE IF se irá a la siguiente línea y comprobará la condición (gbl_cpu_total_util < 20). Si esa condición es true, entonces "The CPU is idling at: " se mostrará en la consola de comandos padv junto con el % de CPU usado.

Instrucción LOOP

Use las instrucciones LOOP para obtener información sobre el sistema. Por ejemplo, se puede averiguar el proceso que usa el mayor porcentaje de CPU o la zona swap que más está siendo utilizada. Esta información puede obtenerse por medio de la instrucción LOOP y las instrucciones pertinentes que usen nombres de métrica respecto a las condiciones del sistema sobre las que se está recopilando información.

Sintaxis:

```
{APPLICATION, APP, CPU, DISK, DISK_DETAIL, FILESYSTEM, FS, FS_DETAIL, LAN, LOGICALVOLUME, LV, LV_DETAIL, NETIF, NFS, NFS_BYSYS_OPS, NFS_OP, PRM, PRM_BYVG, PROCESS, PROC, PROC_FILE, PROC_REGION, PROC_SYSCALL, SWAP, SYSTEMCALL, SC, THREAD, TRANSACTION, TT, TTBIN, TT_CLIENT, TT_INSTANCE, TT_UDM, TT_RESOURCE, TT_INSTANCE_CLIENT, TT_INSTANCE_UDM, TT_CLIENT_UDM, LDOM, PROC_LDOM}
```

LOOP statement

Una instrucción LOOP puede estar anidada en otras instrucciones de sintaxis, pero sólo es posible anidar hasta 5 niveles. La instrucción puede ser una instrucción COMPOUND que contenga un bloque de instrucciones a ser ejecutadas en cada iteración del bucle. Una instrucción BREAK permite el escape de una instrucción LOOP.

Si se tiene una instrucción LOOP en la sintaxis para recopilar determinados datos y no existen datos de métrica correspondientes en el sistema, la secuencia de asesor saltará ese LOOP y continuará en la instrucción de sintaxis siguiente. Por ejemplo, si se ha definido un LOGICAL VOLUME LOOP, pero no se tienen volúmenes lógicos en el sistema, la secuencia de asesor saltará ese LOGICAL VOLUME LOOP y continuará en la instrucción de sintaxis siguiente.

Los bucles que no existan en la plataforma generarán un error de sintaxis.

Mientras la instrucción LOOP itera en cada intervalo, los valores de la métrica usados en la instrucción cambiarán. Por ejemplo, la siguiente instrucción LOOP ejecuta la instrucción PRINT una vez por cada aplicación activa en el sistema, haciendo imprimir el nombre de cada aplicación.

APP LOOP

```
PRINT app_name
```

En un sistema operativo multiproceso como HP_UX 11.23, la secuencias de asesor admiten los THREAD LOOP. Un bucle de subprocesos puede estar anidado dentro de un bucle de procesos con objeto de examinar cada subproceso de un proceso particular. Si se referencia un PROC_metric en un bucle de subprocesos, se pueden obtener resultados inesperados (información sobre subprocesos).

Un bucle de subprocesos puede darse fuera de un bucle de procesos. En este caso examinará todos los subprocesos activos en el sistema. No se deben anidar bucles de procesos en bucles de subprocesos.

Dado que las instrucciones LOOP se inician en cada intervalo, éstas deben usarse con precaución puesto que pueden afectar el rendimiento. Téngase especial cuidado cuando se usen instrucciones LOOP anidadas.

Ejemplo de APPLICATION LOOP

Use la instrucción APPLICATION LOOP para realizar un bucle por todas las aplicaciones activas.

Pueden usarse las métricas globales (gbl_), (tbl_) o de aplicación (app_) con APPLICATION LOOP.

El ejemplo siguiente emplea un bucle de aplicación para detectar la aplicación que tiene el mayor CPU de un intervalo en particular.

```
big_app = ""
highest_cpu = 0
APPLICATION LOOP
  IF (app_cpu_total_util > highest_cpu) THEN
  {
    highest_cpu = app_cpu_total_util
    big_app = app_name
  }
  IF (highest_cpu > 20) THEN
    YELLOW ALERT "The application ", big_app,
      " is the highest CPU user at", highest_cpu, "%"
```

Una vez que encuentra la aplicación, la secuencia de asesor escribe un mensaje en la consola de comandos padv con app_name y el valor de CPU, si el valor de CPU es superior a 20.

Ejemplo de CPU LOOP

Use la instrucción CPU LOOP para realizar un bucle por datos sobre el uso de CPU del sistema. Pueden usarse las métricas globales (gbl_) (tbl_) o por CPU (bycpu_) con CPU LOOP.

El siguiente ejemplo imprime el porcentaje de uso de CPU correspondiente a cada CPU del sistema.

```
Print "-----", glb_stattime, "-----"
```

Ejemplo de

```
PRINT "CPU # ", bycpu_id, " used ", bycpu_cpu_total_util, " % CPU"
```

En un sistema con dos CPU, el resultado de dos intervalos será:

```
-----10:52:01-----
CPU #          0 used    0,6 % CPU
CPU #          1 used    3,4 % CPU
-----10:52:11-----
CPU #          0 used    0,4 % CPU
CPU #          1 used    2.3 % CPU
```

Ejemplo de DISK LOOP

Use la instrucción DISK LOOP para realizar un bucle por los dispositivos de disco configurados. Al usar este bucle, la secuencia de asesor comprueba información sobre disco específica que se muestra en la ventana IO by Disk. Pueden usarse las métricas globales (gbl_) (tbl_) o por disco con DISK LOOP.

El siguiente ejemplo imprime la frecuencia de escritura física correspondiente a cada disco del sistema.

```
PRINT "-----", gbl_stattime, "-----"
DISK LOOP
```

```
    PRINT bydisk_devname, " write rate: ", bydisk_phys_write_rate
```

En un sistema con tres discos, el resultado de dos intervalos será:

```
-----11:00:23-----
```

```
/dev/hdisk0      write rate:    2.4
/dev/hdisk1      write rate:    0.0
/dev/cd0         write rate:    0.0
```

```
-----11:00:33-----
```

```
/dev/hdisk0      write rate:    0.0
/dev/hdisk1      write rate:    0.0
/dev/cd0         write rate:    0.0
```

Ejemplo de FILE SYSTEM LOOP

FILE SYSTEM LOOP ha sido diseñado para realizar un bucle por los sistemas de archivos configurados y permitir a la secuencia de asesor notificar sobre información accesible en la ventana IO By File System. Pueden usarse las métricas globales (gbl_), (tbl_) o métricas de sistema de archivos (fs_) con FILE SYSTEM LOOP.

El ejemplo siguiente indica el espacio utilizado por cada dispositivo de sistema de archivo de un sistema con tres dispositivos.

```
PRINT "-----", gbl_stattime, "-----"
FS LOOP
```

```
    PRINT fs_devname, " is ", fs_space_util, "% full at ",
          fs_max_size, " megabytes"
```

El resultado de dos intervalos en un sistema con tres sistemas de archivo es:

```
-----11:11:28-----
```

```
/dev/hd4         is  77.9% full at    32 megabytes
/dev/hd2         is  94.9% full at   928 megabytes
/dev/hd9var      is  93,9% full at    56 megabytes
```

```
-----11:11:38-----
```

```
/dev/hd4         is  77.9% full at    32 megabytes
/dev/hd2         is  94.9% full at   928 megabytes
/dev/hd9var      is  93.6% full at    56 megabytes
```

Ejemplo de NFS BY OPERATION LOOP

Use NFS BY OPERATION LOOP para realizar un bucle por las operaciones NFS realizadas. Al usar este LOOP, la secuencia de asesor busca operaciones NFS específicas. Pueden usarse las métricas globales (gbl_), (tbl_) o por operación con NFS_OP LOOP.

El ejemplo siguiente imprime las operaciones de cliente y servidor realizadas:

```
PRINT "-----", gbl_stattime, "-----"
NFS_OP LOOP
    PRINT byop_server_count," server and ",byop_client_count,
        " client ",byop_name," operations performed"
```

En un sistema que no realice ninguna actividad como servidor NFS pero con usuarios que hagan listas de directorio en otro servidor NFS, el resultado será:

```
-----14:55:41-----
    0 server and      0 client null          operations performed
    0 server and      2 client getattr       operations performed
    0 server and      0 client setattr       operations performed
    0 server and      0 client root          operations performed
    0 server and     886 client lookup      operations performed
    0 server and     884 client readlink    operations performed
    0 server and      0 client read          operations performed
    0 server and      0 client writecache    operations performed
    0 server and      0 client write         operations performed
    0 server and      0 client create        operations performed
    0 server and      0 client remove        operations performed
    0 server and      0 client rename        operations performed
    0 server and      0 client link          operations performed
    0 server and      0 client symlink       operations performed
    0 server and      0 client mkdir         operations performed
    0 server and      0 client rmdir         operations performed
    0 server and     28 client readdir     operations performed
    0 server and      1 client statfs        operations performed
```

Ejemplo de NETWORK INTERFACE LOOP

Use NETWORK INTERFACE LOOP para realizar un bucle por dispositivos LAN configurados.

```
# This version will only work with hp-ux 11.x.If you want it to
# work for 10.20 you need to remove the "BYNETIF_QUEUE," string
# below as that metric is only available from 11.x glance.
# The following string variable should be changed to the interface
# of interest. Por ejemplo:
#   netif_to_examine = "lan0"
# If you want to see all interfaces, leave it an empty string (""):
#   netif_to_examine = ""
# initialize variables:

headers_printed = headers_printed
netif loop {
# print information for the selected interface or if null then all:
IF (BYNETIF_NAME == netif_to_examine) or
    (netif_to_examine == "") THEN
    {

# print headers the first time through the loop:
        IF headers_printed == 0 THEN
            {
                print "Time    Interface  InPkts OutPkts  OutQ  Colls  Errs"
                print "  "
                headers_printed = 1
            }
# print one line per interface reported:
        print GBL_STATTIME, " ", BYNETIF_NAME|8,
            BYNETIF_IN_PACKET, BYNETIF_OUT_PACKET,
            BYNETIF_QUEUE, BYNETIF_COLLISION, BYNETIF_ERROR
        # (note that some interface types do not report collisions or
        # errors)
    }
}
print "  "
```

El resultado es:

Time	Interface	InPkts	OutPkts	OutQ	Colls	Errs
22:43:42	lan3	49	3	0	0	0
22:43:42	lan0	0	0	0	0	0
22:43:42	lan1	0	0	0	0	0
22:43:42	lan2	0	0	0	0	0
22:43:42	lo0	0	0	0	0	0
22:43:47	lan3	329	2	0	0	0
22:43:47	lan0	0	0	0	0	0
22:43:47	lan1	0	0	0	0	0
22:43:47	lan2	0	0	0	0	0
22:43:47	lo0	0	0	0	0	0

Ejemplo de LOGICAL VOLUME

Use bucles LOGICAL VOLUME para realizar un bucle por volúmenes lógicos configurados. Pueden usarse las métricas globales (gbl_), (tbl_) o métricas de volumen lógico con LOGICAL VOLUME LOOP.

```
PRINT "-----", gbl_stattime, "-----"
LV LOOP
PRINT "Volume ", lv_dirname, " was read at a rate of ",
lv_read_rate, " per second"
```

El resultado de dos intervalos en un sistema con volúmenes lógicos:

```
-----11:46:50-----
Volume /dev/vg00          was read at a rate of    0.0 per second
Volume /dev/vg00/group   was read at a rate of    0.0 per second
Volume /dev/vg00/1vol3   was read at a rate of  314,3 per second

-----11:47:00-----
Volume /dev/vg00          was read at a rate of    0.0 per second
Volume /dev/vg00/group   was read at a rate of    0.0 per second
Volume /dev/vg00/1vol3   was read at a rate of   70.6 per second
```

Ejemplo de PRM LOOP

Sólo HP-UX. Use PRM LOOP para realizar un bucle por información procedente de los grupos Process Resource Manager (PRM). Pueden usarse las métricas globales (gbl_), (tbl_) o de aplicación (app_) con PRM LOOP.

El siguiente ejemplo de PRM LOOP comprueba si hay colas de ejecución altas y grupos PRM que excedan sus derechos de CPU.

```
IF gbl_run_queue > 3 THEN {
    print " "

    print "--- High run queue = ", gbl_run_queue, " at ", gbl_stattime,
        " ---"

    prm loop {
        IF app_prm_state > 2 THEN
            IF app_cpu_total_util > app_prm_cpu_entitlement THEN
                print "    Note PRM group ", app_name_prm_groupname,
                    " exceeds entitlement."
            }
        }
    }
}
```

El resultado impreso en cada intervalo es:

```
--- High run queue =   3.4 at 15:53:29 ---
    Note PRM group Testing exceeds entitlement.
```

Ejemplo de PRM_BYVG LOOP

Sólo HP-UX. Use el bucle PRM_BYVG para realizar un bucle por grupos PRM de un grupo de volumen. (Adviértase que la información de PRM sólo está disponible respecto a grupos de volumen especificados en el archivo de configuración PRM). El bucle PRM_BYVG deberá estar anidado en un bucle LV. El siguiente ejemplo muestra las estadísticas de uso de recursos del disco por grupo PRM.

```
PRM loop {
    disk_state = app_prm_disk_state
}

IF disk_state == 0 THEN{
    print "    Disk manager state: Not Installed"
}

else IF disk_state == 1 THEN {
    print "    Disk manager state: Reset"
}
```



```

else IF disk_state == 2 THEN {
    print "  Disk manager state: Disabled"
}

else IF disk_state == 3 THEN {

    print "  Disk manager state: Enabled"

    lv loop {

        IF lv_type == "G" THEN {

            print "  Volume Group: ", lv_dirname
            print "                %          %          KB"
            print "PRM Group      PRMID entitled achieved  transferred"
            print "-----"

            prm_byvg loop {
                print prm_byvg_prm_groupname|13, prm_byvg_prm_groupid|5,
                    prm_byvg_group_entitlement|8, prm_byvg_group_util|8,
                    prm_byvg_transfer
            }
            print "  "
        }
    }
}
}

```

El resultado en cada intervalo es:

Disk manager state: Enabled

Volume Group: /dev/vg00

	%	%	KB	
PRM Group	PRMID	entitled	achieved	transferred

PRM_SYS	0	0	100	8
OTHERS	1	50	0	0
tools	2	50	0	0

Ejemplo de PROCESS LOOP

Use la instrucción PROCESS LOOP para realizar un bucle por todos los procesos activos. Pueden usarse las métricas globales (gbl_), (tbl_) o de proceso (proc_) con PROCESS LOOP. El ejemplo siguiente usa un PROCESS LOOP para detectar el proceso que presenta el CPU más alto de un intervalo en particular.

```
big_proc_id = 0
big_proc_name = ""
big_proc_cpu = 0
PROCESS LOOP
IF proc_cpu_total_util > big_proc_cpu THEN {
    big_proc_cpu = proc_cpu_total_util
    big_proc_name = proc_proc_name
    big_proc_id = proc_proc_id
}

IF big_proc_cpu > 10 THEN
    YELLOW ALERT "Possible loop, process ", big_proc_name,
        " pid ", big_proc_id|6|0, " using ", big_proc_cpu, " % CPU"
```

Ejemplo de SWAP LOOP

Use SWAP LOOP to LOOP en las zonas swap configuradas y permita a la secuencia de asesor notificar sobre información procedente de la ventana Swap Space. Pueden usarse las métricas globales (gbl_), (tbl_) o de intercambio (byswp_) con SWAP LOOP.

El siguiente ejemplo informa sobre el espacio swap disponible en un sistema con dos dispositivos swap.

```
PRINT "-----", gbl_stattime, "-----"
SWAP LOOP
    PRINT BYSWP_SWAP_SPACE_NAME, " has ", BYSWP_SWAP_SPACE_USED,
        " used out of", BYSWP_SWAP_SPACE_AVAIL, " megabytes "
```

En un sistema con una zona swap, el resultado impreso correspondiente a dos intervalos es el siguiente:

```
-----15:31:59-----
/dev/hd6          has      37  used out of 128 megabytes

-----15:32:09-----
/dev/hd6          has      37  used out of 128 megabytes
```

Ejemplo de SYSTEM CALL LOOP

Use SYSTEM CALL LOOP para realizar un bucle por las llamadas del sistema. Al usar SYSTEM CALL LOOP, la secuencia de asesor comprueba la información disponible en la ventana System Call. Pueden usarse las métricas globales (gbl_) (tbl_) o de llamada del sistema (syscall_) con SYSTEM CALL LOOP.

El siguiente ejemplo comprueba si hay frecuencia de llamadas del sistema altas y, a continuación, imprime la llamada más frecuente.

```
IF gbl_syscall_rate > 6000 THEN {
    print " "
    print "--- High syscall rate = ", gbl_syscall_rate, " at ",
        gbl_stattime, " ---"
    highestrate = 0

    syscall loop {
        IF syscall_call_rate > highestrate THEN {
            highestrate = syscall_call_rate
            highestcall = syscall_call_name
        }
    }
    print " Most frequent syscall was ", highestcall, " at",
        highestrate, " per second"
}
```

El resultado es:

```
--- High syscall rate = 6750.6 at 15:50:27 ---
```

```
Most frequent syscall was gettimeofday at 6632.90 per second
```

Ejemplo de TT LOOP

Use TT LOOP para realizar un bucle por la información sobre transacciones que ha sido registrada durante el último intervalo. Al usar este bucle, la secuencia de asesor comprueba información sobre transacción específica que se muestra en la ventana Transaction Tracking. Pueden usarse las métricas globales (gbl_), (tbl_) o de rastreo de transacción (tt_) con TT LOOP.

El ejemplo siguiente imprime el número de transacciones finalizadas y el tiempo de respuesta promedio correspondiente a cada nombre de transacción registrada del sistema.

```
PRINT "-----", gbl_stattime, "-----"
TT LOOP
PRINT tt_name, " had ", tt_count, " transactions; ",
      "response time ", tt_wall_time_per_tran, " secs"
```

En un sistema con cuatro transacciones el resultado de dos intervalos es el siguiente:

```
-----13:24:44-----
First_Transaction had 1 transactions; response time 1,000355 secs
Second_Transaction had 1 transactions; response time 2,000221 secs
Third_Transaction had 1 transactions; response time 3,000231 secs
Fourth_Transaction had 0 transactions; response time 0.000000 secs

-----13:24:54-----
First_Transaction had 3 transactions; response time 1.000383 secs
Second_Transaction had 1 transactions; response time 2.000216 secs
Third_Transaction had 0 transactions; response time 0.000000 secs
Fourth_Transaction had 0 transactions; response time 0.000000 secs
```

Ejemplo de TTBIN LOOP

Use TTBIN LOOP, que deberá estar anidado a un bucle TT, para realizar un bucle por los contenedores de tiempo de respuesta de cada transacción activa del sistema. Al usar este bucle, la secuencia de asesor comprueba información sobre transacción específica que se muestra en la ventana Transaction Graph. Pueden usarse las métricas globales (gbl_), (tbl_), de rastreo de transacción o de bin de rastreo de transacción con TTBIN LOOP.

El siguiente ejemplo imprime los contenedores de tiempo de respuesta correspondientes a cada nombre de transacción que han tenido transacciones finalizadas durante el intervalo.

```
PRINT "-----", gbl_stattime, "-----"
TT LOOP
  IF (tt_count > 0) THEN
  {
    print "Transaction ", tt_name, " had ", tt_count, " transactions"
    lower_bin_limit = 0
    TTBIN LOOP
    {
      IF (ttbin_trans_count > 0) THEN {
        print " ", ttbin_trans_count, " were between ",
          lower_bin_limit, " and ", ttbin_upper_range, " seconds"
        lower_bin_limit = ttbin_upper_range
      }
    }
  }
}
```

En un sistema con cuatro transacciones el resultado impreso de dos intervalos es el siguiente:

```
-----13:46:31-----
Transaction First_Transaction  had      4 transactions
      2 were between      1,00 and  2,000000 seconds
Transaction Second_Transaction  had      1 transactions
      1 were between      2,00 and  3,000000 seconds
Transaction Third_Transaction   had      1 transactions
      1 were between      3.00 and  5.000000 seconds

-----13:46:41-----
Transaction First_Transaction  had      3 transactions
      1 were between      1,00 and  2,000000 seconds
Transaction Second_Transaction  had      1 transactions
      1 were between      2,00 and  3,000000 seconds
Transaction Fourth_Transaction  had      1 transactions
      1 were between      3.00 and  5.000000 seconds
```

Ejemplo de TT LOOP ARM

Con ARM 2.0, los bucles TT_CLIENT, TT_INSTANCE y TT_UDM pueden estar anidados en un TT LOOP. El bucle TT_CLIENT indica las transacciones correlacionadas, el bucle TT_INSTANCE enumera hasta 2048 instancias de transacciones y el bucle TT_UDM enumera las mediciones de usuario correspondientes a una transacción dada. Pueden usarse las métricas globales (gbl_), (tbl_) o de rastreo de transacción con TT LOOP.

El ejemplo que sigue a continuación muestra cómo usar bucles múltiples para ver mediciones de usuario correspondientes a una instancia de transacción dada.

Ejemplo 1: Buscar violaciones SLO

```
# The following example loops through all transactions looking for
# SLO violations, then prints the UDM information for all
# instances:
```

```
print "-----", GBL_STATTIME, "-----"
```

```
tt loop {
```

```
  IF tt_slo_count > 0 THEN {
```

```
    print " "
```

```
    print "SLO violation count:", tt_slo_count,
```

```
          " for transaction:", tt_name, " user:", tt_uname,
```

```
          " app:", tt_app_name, " threshold: ", tt_slo_threshold
```

```
    tt_instance loop {
```

```
      starttime = gbl_stattime - gbl_interval
```

```
      IF tt_instance_stop_time > starttime THEN {
```

```
        # found a completed instance in the transaction, print info:
```

```
        print "instance pid:", tt_instance_proc_id,
```

```
              " wall time:", tt_instance_wall_time
```

```
        tt_instance_udm loop {
```

```
          print "    ", tt_instance_user_measurement_name|44,
```

```
                " value= ", tt_instance_user_measurement_value
```

```
        }
```

```
      }
```

```
    }
```

```
  }
```

```
}
```

A continuación se muestra el resultado de un intervalo.

-----17:19:03-----
SLO violation count: 1 for transaction:Client_tra00 user:gracel
app:Client_App10 threshold: 5.000000
instance pid: 12137 wall time: 13.0407

SLO violation count: 1 for transaction:Server_transaction user:joe
app:Server_Application threshold: 5.000000
instance pid: 12137 wall time: 13.0358

Metric #1 - Type 1 is a COUNTER32	value=	32
Metric #2 - Type 4 is a GAUGE32	value=	37
Metric #3 - Type 2 is a COUNTER64	value=	19088743
Metric #4 - Type 9 is a STRING8	value=	String 8
Metric #5 - Type 3 is a COUNTER32/DIVISOR32	value=	2.000
Metric #6 - Type 8 is a NUMERICID64	value=	19088434
The last field is always a STRING32	value=	0

instance pid: 12137 wall time: 3.0291

Metric #1 - Type 1 is a COUNTER32	value=	32
Metric #2 - Type 4 is a GAUGE32	value=	37
Metric #3 - Type 2 is a COUNTER64	value=	19088743
Metric #4 - Type 9 is a STRING8	value=	String 8
Metric #5 - Type 3 is a COUNTER32/DIVISOR32	value=	21.333
Metric #6 - Type 8 is a NUMERICID64	value=	19088434
The last field is always a STRING32	value=	0

instance pid: 12137 wall time: 3.0256

Metric #1 - Type 1 is a COUNTER32	value=	32
Metric #2 - Type 4 is a GAUGE32	value=	37
Metric #3 - Type 2 is a COUNTER64	value=	19088743
Metric #4 - Type 9 is a STRING8	value=	String 8
Metric #5 - Type 3 is a COUNTER32/DIVISOR32	value=	21.333
Metric #6 - Type 8 is a NUMERICID64	value=	19088434
The last field is always a STRING32	value=	0

instance pid: 12137 wall time: 2.0201

Metric #1 - Type 1 is a COUNTER32	value=	32
Metric #2 - Type 4 is a GAUGE32	value=	37
Metric #3 - Type 2 is a COUNTER64	value=	19088743
Metric #4 - Type 9 is a STRING8	value=	String 8
Metric #5 - Type 3 is a COUNTER32/DIVISOR32	value=	21.333

```

Metric #6 - Type 8 is a NUMERICID64      value=      19088434
The last field is always a STRING32      value=          0
instance pid: 12137 wall time:  1.0101
Metric #1 - Type 1 is a COUNTER32        value=          32
Metric #2 - Type 4 is a GAUGE32          value=          37
Metric #3 - Type 2 is a COUNTER64        value=      19088743
Metric #4 - Type 9 is a STRING8          value=      String 8
Metric #5 - Type 3 is a COUNTER32/DIVISOR32 value=        21.333
Metric #6 - Type 8 is a NUMERICID64      value=      19088434
The last field is always a STRING32      value=          0

```

Ejemplo 2: Sintaxis ARM 2.0

```

# The following example prints info for all completed transactions
# during the interval.

```

```

print "-----", GBL_STATTIME, "-----"

header_printed = 0
tt loop {

    tt_instance loop {

        starttime = GBL_STATTIME - GBL_INTERVAL

        IF TT_INSTANCE_STOP_TIME > starttime THEN {

            IF header_printed == 0 THEN {
                print " "
                print "TranID  StartTime                StopTime",
                    "                "
                header_printed = 1
            }

            print TT_TRAN_ID|6, " ", TT_INSTANCE_START_TIME, " ",
                TT_INSTANCE_STOP_TIME
            print "                TranName: ", TT_NAME|40
        }
    }
}

```



```
}  
}
```

A continuación se muestra el resultado de un intervalo.

-----17:21:24-----

TranID	StartTime	StopTime
3	Wed Jun 3 17:21:07 1998	Wed Jun 3 17:21:20 1998
	TranName: Client_tra00	
7	Wed Jun 3 17:21:07 1998	Wed Jun 3 17:21:20 1998
	TranName: Server_transaction	
7	Wed Jun 3 17:21:17 1998	Wed Jun 3 17:21:20 1998
	TranName: Server_transaction	
7	Wed Jun 3 17:21:17 1998	Wed Jun 3 17:21:20 1998
	TranName: Server_transaction	
7	Wed Jun 3 17:21:18 1998	Wed Jun 3 17:21:20 1998
	TranName: Server_transaction	
7	Wed Jun 3 17:21:19 1998	Wed Jun 3 17:21:20 1998
	TranName: Server_transaction	

Ejemplo de bucle LDOM

Sólo HP-UX 11iv2 y superior. Use la instrucción de bucle LDOM para realizar un bucle por los datos del dominio de localidad del sistema. Pueden usarse las métricas globales (gbl_) (tbl_) o métricas de dominio de localidad (ldom_) con LDOM LOOP.

El ejemplo siguiente imprime el uso de memoria LDOM de cada dominio de localidad del sistema.

```
print " "
PRINT "-----", gbl_stattime, "-----"
print "LDOM Phys          Num          Mem      Mem      Mem"
print "  ID   ID Active CPUs Type        Avail      Free Used %"
PRINT "-----"
LDOM LOOP
{
  print LDOM_ID, " ", LDOM_PHYS_ID, " ", LDOM_ACTIVE, " ", LDOM_NUM_CPU, " ",
  LDOM_MEM_TYPE, " ", LDOM_MEM_AVAIL, " ", LDOM_MEM_FREE, " ", LDOM_MEM_UTIL
}
PRINT "-----"
```

En un sistema con tres LDOM, el resultado de dos intervalos es:

```
-----03:30:27-----
LDOM Phys          Num          Mem      Mem      Mem
  ID   ID Active CPUs Type        Avail      Free Used %
-----
   0   0     1    4 LOCAL        0mb       0mb     0.0
   1   2     1    4 LOCAL        0mb       0mb     0.0
  na  na     1    0 GLOBAL      8.0gb     5.5gb   30.5
-----
```

```
-----03:30:32-----
LDOM Phys          Num          Mem      Mem      Mem
  ID   ID Active CPUs Type        Avail      Free Used %
-----
   0   0     1    4 LOCAL        0mb       0mb     0.0
   1   2     1    4 LOCAL        0mb       0mb     0.0
  na  na     1    0 GLOBAL      8.0gb     5.5gb   30.5
-----
```

Ejemplo de PROC_LDOM LOOP

Sólo HP-UX 11iv2 y superior. Use el bucle PROC_LDOM para hacer un bucle por los dominios de localidad de los que el proceso puede obtener memoria.

El bucle PROC_LDOM deberá estar anidado en un bucle PROCESS.

El siguiente ejemplo muestra el proceso scopeux que puede obtener memoria de diferentes LDOM:

```
print " "
PRINT "-----", gbl_stattime,
"-----"

print "Process          LDOM LDOM          RSS          RSS          RSS          RSS
LDOM"

print "Name              ID Type          Total          Shared          Private          Weighted
Mem %"

PRINT
"-----"
-----"

PROCESS LOOP
{
if PROC_PROC_NAME == "scopeux" then
{
PROC_LDOM LOOP
{
print PROC_PROC_NAME, " ", PROC_LDOM_ID, " ", PROC_LDOM_TYPE, " ",
PROC_LDOM_TOTAL, " ", PROC_LDOM_SHARED, " ", PROC_LDOM_PRIVATE, " ",
PROC_LDOM_WEIGHTED, " ", PROC_LDOM_PCT
}
}
}

PRINT
"-----"
-----"
```

Resultado de dos intervalos en un sistema con tres LDOM filtrados para el proceso scopeux:

```
-----04:53:50-----
----
Process          LDOM LDOM          RSS          RSS          RSS          RSS          LDOM
Name             ID  Type          Total        Shared       Private       Weighted     Mem %
-----
scopeux          0  LOCAL          0kb          0kb          0kb          0kb          0.0
scopeux          1  LOCAL          0kb          0kb          0kb          0kb          0.0
scopeux          na GLOBAL        31.9mb       23.7mb       8.2mb        13.4mb       0.2
-----
```

```
-----04:53:55-----
----
Process          LDOM LDOM          RSS          RSS          RSS          RSS          LDOM
Name             ID  Type          Total        Shared       Private       Weighted     Mem %
-----
scopeux          0  LOCAL          0kb          0kb          0kb          0kb          0.0
scopeux          1  LOCAL          0kb          0kb          0kb          0kb          0.0
scopeux          na GLOBAL        31.9mb       23.7mb       8.2mb        13.4mb       0.2
-----
```

Instrucción PRINT

Use la instrucción PRINT para imprimir en stdout (la consola de comandos `padv`) los datos que están siendo recopilados. El usuario puede usar la instrucción PRINT para registrar métrica o variables calculadas.

Sintaxis:

```
PRINT printlist
```

```
PRINT Example
```

```
PRINT "The Application OTHER has a total CPU of ",
```

```
    other:app_cpu_total_util, "%"
```

A ser invocado, esta instrucción imprime un mensaje en la consola de comandos `padv` similar al siguiente:

La aplicación OTHER tiene un CPU total de 89%

Instrucción SYMPTOM

Sintaxis:

```
SYMPTOM variable [TYPE = {CPU, DISK, MEMORY, NETWORK}]
RULE measurement {>, <, >=, <=, ==, !=} value PROB probability
[RULE measurement {>, <, >=, <=, ==, !=} value PROB probability]
.
.
.
```

Las palabras clave SYMPTOM y RULE son exclusivas de la instrucción SYMPTOM y no pueden ser usadas en otras instrucciones de sintaxis. La instrucción SYMPTOM debe ser una instrucción de nivel superior y no puede ser anidada en otra instrucción.

variable es un nombre de variable que será el nombre de este síntoma. Los nombres de variables definidos en la instrucción SYMPTOM pueden ser usados en otras instrucciones de sintaxis, pero el valor de variable no deberá ser cambiado en esas instrucciones.

RULE es una opción de la instrucción SYMPTOM y no puede ser usada independientemente. En la instrucción SYMPTOM se pueden usar tantas opciones RULE como se necesiten.

La variable SYMPTOM es evaluada de acuerdo a las reglas en cada intervalo.

- Measurement es el nombre de una variable o métrica que es evaluada como parte de la RULE.
- Value es una constante, variable o métrica que es comparada con la medición.
- Probability es una métrica, variable o constante numérica.

Las probabilidades de todas las SYMPTOM RULE que son true son agregadas conjuntamente para crear un valor SYMPTOM. El valor SYMPTOM aparece entonces en el mensaje de la consola de comandos padv.

La suma de todas las probabilidades en las que la condición entre medición (measurement) y valor (value) es true es la probabilidad de que ocurra el síntoma.

Ejemplo de SYMPTOM

Sintaxis:

```
SYMPTOM CPU_Bottleneck TYPE=CPU
RULE gbl_cpu_total_util > 50  PROB 25
RULE gbl_cpu_total_util > 85  PROB 25
RULE gbl_cpu_total_util > 90  PROB 25
RULE gbl_run_queue      > 3   PROB 50
SYMPTOM CPU_Level TYPE=CPU
RULE gbl_cpu_sys_mode_util > 40  PROB 25
RULE gbl_cpu_sys_mode_util > 50  PROB 25
RULE gbl_cpu_sys_mode_util > 60  PROB 25
RULE gbl_cpu_sys_mode_util > 70  PROB 50
```

El síntoma CPU definido anteriormente que presente la probabilidad (PROB) total más alta, será el síntoma que determine el nivel de gravedad del mensaje en la consola de comandos `padv`.

Ejemplo de SYMPTOM: Global CPU Bottleneck (Cuello de botella CPU global)

```
SYMPTOM Symp_Global_Cpu_Bottleneck TYPE=CPU
RULE gbl_cpu_total_util > 50  PROB 25
RULE gbl_cpu_total_util > 85  PROB 25
RULE gbl_cpu_total_util > 90  PROB 25
RULE gbl_run_queue      > 3   PROB 75
```

Por ejemplo, si la utilización de CPU (`gbl_cpu_total_util`) correspondiente al intervalo era del 93% y la cola de ejecución era 2, las primeras tres reglas serían true de manera que se agregaría tres veces 25 a la probabilidad. Puesto que la cuarta regla no era true, no se agregaría 75. Por lo tanto, la variable `Symp_Global_Cpu_Bottleneck` tendría un valor de 75 (por ciento) en ese intervalo.

11 Uso de Performance Collection Component en Windows

Para acceder a la interfaz gráfica de usuario de Componente Performance Collection, haga clic en el icono de HP Operations Agent Software que se encuentra en la carpeta siguiente:

Inicio→Programas→HP→Operations Agent→Performance Collection Component

Figura 2 Ventana principal de Performance Collection Component



En este capítulo se describen las tareas siguientes que se pueden realizar con la interfaz gráfica de Componente Performance Collection:

- Tipos y clases de datos
- Niveles de resumen
- Rangos de los datos que se van a extraer o exportar
- Extracción de datos de archivos de registro y Exportación de datos de archivos de registro
- Almacenamiento de datos de archivos de registro
- Cambio de tamaño de un archivo de registro
- Exploración de un archivo de registro para obtener más información
- Análisis de un archivo de registro
- Configuración de plantillas de exportación
- Configuración de opciones de usuario
- Configuración de los parámetros de recopilación
- Configuración de definiciones de alarma
- Comprobación del estado de Componente Performance Collection
- Creación de recopilación de contadores de rendimiento

Antes de comenzar a usar Componente Performance Collection en aquellas tareas que impliquen la extracción, exportación o archivado de datos, lea las secciones siguientes. En estas secciones se describe la selección de tipos de datos, los niveles de resumen y los rangos de datos que se van a extraer, exportar o archivar.

Tipos y clases de datos

Se pueden seleccionar los siguientes tipos de datos del archivo de registro `scopent` para su extracción o exportación:

Tipos de datos	Tipo de medida
<code>global</code>	información global o de todo el sistema.
<code>application</code>	procesos de las aplicaciones definidas por el usuario
<code>configuration</code>	uso de configuración del sistema
<code>process</code>	procesos interesantes seleccionados
<code>disk</code>	uso de dispositivos de disco
<code>filesystem</code>	uso de discos lógicos
<code>logicalsystem</code>	uso del sistema lógico
<code>cpu</code>	uso de CPU
<code>netif</code>	uso de dispositivos de la interfaz de red
<code>transaction</code>	datos de seguimiento de transacciones

Se pueden seleccionar los datos del archivo de registro DSI para su exportación de acuerdo con la clase. Cada clase representa un origen de datos entrantes y consta de un grupo de elementos de datos relacionados (métricas) que se registran juntos.

Niveles de resumen

Para exportar los datos, hay que especificar el nivel de resumen que se desea (detalle, resumen o ambos) cuando se exportan los datos del archivo de registro.

- El nivel de detalle especifica que se exportan los datos de detalles en un período de 5 minutos en todos los tipos de datos, excepto los datos de proceso, en los que se exportan los datos de detalles en un período de 1 minuto.
- El nivel de resumen indica que se exportan los datos resumidos en un período de 1 hora.
- El nivel de detalle y el de resumen juntos proporcionan la máxima cantidad de datos exportados.

El proceso de resumen afecta al tamaño de los datos exportados. Por ejemplo, los datos de resumen de cada hora representan aproximadamente una décima parte del tamaño de datos de detalles de 5 minutos.

Rangos de los datos que se van a extraer o exportar

Se pueden seleccionar datos específicos para extraer o exportar, dependiendo de la fecha y hora en que fueron registrados. Por ejemplo, es posible que se desee que los datos se registren cada día (de lunes a domingo), de 8:00 a.m. a 8:00 p.m. durante un período de 30 días, comenzando en una fecha determinada.

Si no se especifica un rango específico de datos para exportar, los datos se extraen o se exportan usando la fecha de inicio predeterminada, es decir, 30 días antes de la última fecha del archivo de registro. O, si hay menos de 30 días con datos, la fecha del primer registro del archivo de registro.

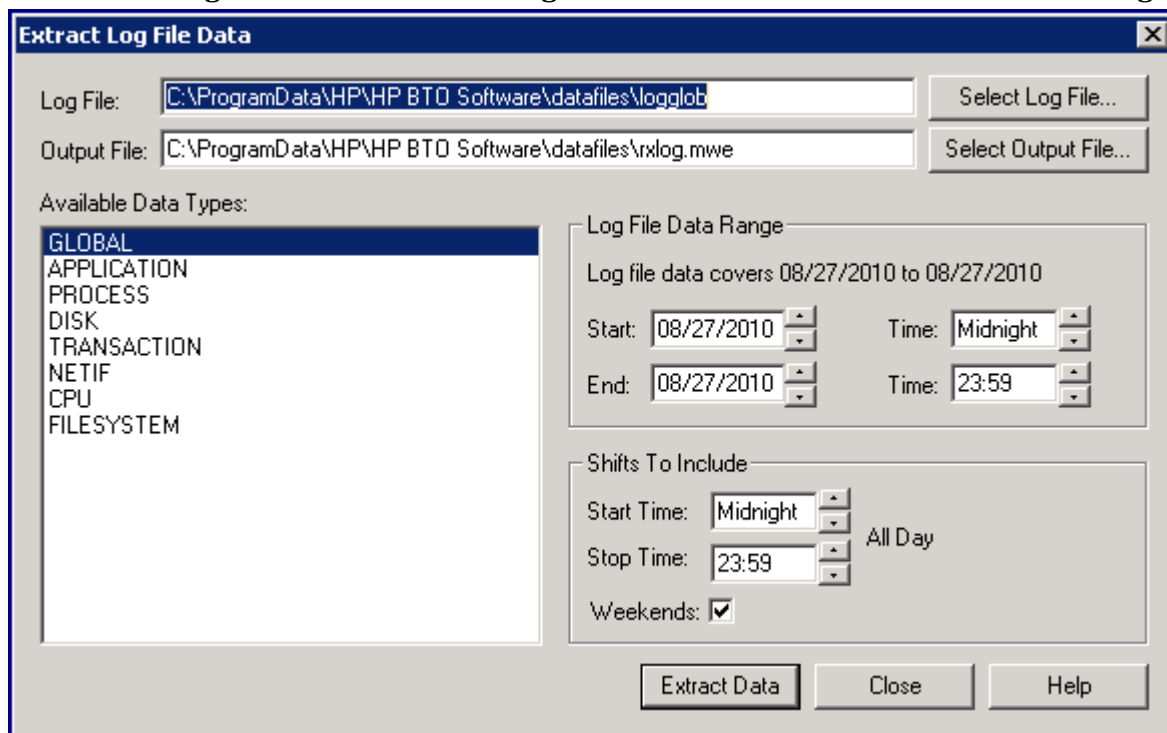
También se puede limitar la extracción o exportación de datos registrados durante determinadas horas del día que se correspondan a turnos de trabajo (como 8:00 a.m. a 5:00 p.m.) de lunes a viernes. Si no se especifica ningún turno, el valor predeterminado es un valor de 24 horas para la extracción o exportación de datos para cada día, incluidos los fines de semana.

Extracción de datos de archivos de registro

El recopilador de datos, *scopent*, recopila datos continuamente y los registra en los archivos de registro sin procesar. Puede extraer datos específicos del conjunto predeterminado de archivos de registro global, *logglob*, en archivos de registro extraídos que se pueden usar posteriormente para que programas de análisis, como HP Performance Manager, los archiven o analicen. También se pueden extraer datos de los archivos de registro extraídos existentes. *No se pueden* extraer datos de archivos de registro DSI.

Cuando se especifica *logglob*, se abrirán automáticamente todos los demás archivos de registro sin procesar del conjunto de archivos de registro. Por ejemplo, no es necesario abrir el archivo de registro *logproc* para extraer datos de proceso; si se abre *logglob*, se podrá acceder a todos los tipos de datos en el conjunto de archivos de registro sin procesar.

Figura 3 Cuadro de diálogo de extracción de datos de archivos de registro



- Si aparecen los cuadros de diálogo *Extract Log File Data* o *Export Log File Data* después de iniciar *Componente Performance Collection*, aparecerá el nombre del archivo de registro global predeterminado, *logglob*, en el cuadro *Log File* para indicar el archivo de registro actualmente activo. *Logglob* sigue activo hasta que se cierre *Componente Performance Collection* o se seleccione un archivo de registro diferente. Si se selecciona otro archivo de registro, el nombre de ese archivo aparecerá en el cuadro *Log File* como archivo de registro actualmente activo.

Para extraer datos de archivos de registro

Para extraer datos de archivos de registro, realice los pasos siguientes:

- 1 Haga clic en **Extract** en el menú **Logfile** en la ventana principal. Aparecerá el cuadro de diálogo **Extract Log File Data**, mostrando los nombres del archivo de registro actualmente activo y el archivo de salida actualmente seleccionado. Se puede especificar fácilmente un archivo de registro y un archivo de salida diferentes.
- 2 Después de seleccionar el tipo de datos que se van a extraer, se selecciona el rango de datos del archivo de registro que se van a extraer y los turnos que se van a incluir. Haga clic en el botón **Extract Data** para iniciar el proceso de registro.

Para obtener instrucciones detalladas sobre cómo extraer datos de archivos de registro, elija **Help Topics** en el menú de ayuda, seleccione "**How Do I...?**," y, a continuación, seleccione "**Extract log file data**".

Exportación de datos de archivos de registro

Se pueden exportar datos específicos de archivos de registro *scopent* extraídos o sin procesar, o desde archivos de registros DSI en archivos de exportación con formato que pueden utilizar hojas de cálculo y otras herramientas de generación de informes.

La función de exportación *no* suprime ningún dato del archivo de registro.

En las siguientes secciones se trata lo siguiente:

- Atributos de archivos que se pueden especificar para los datos exportados
- Plantillas de exportación que se pueden utilizar para simplificar la tarea de exportación
- Archivo de exportación predeterminado
- Introducción general de la tarea de exportación

Figura 4 Cuadro de diálogo **Export Log File Data**

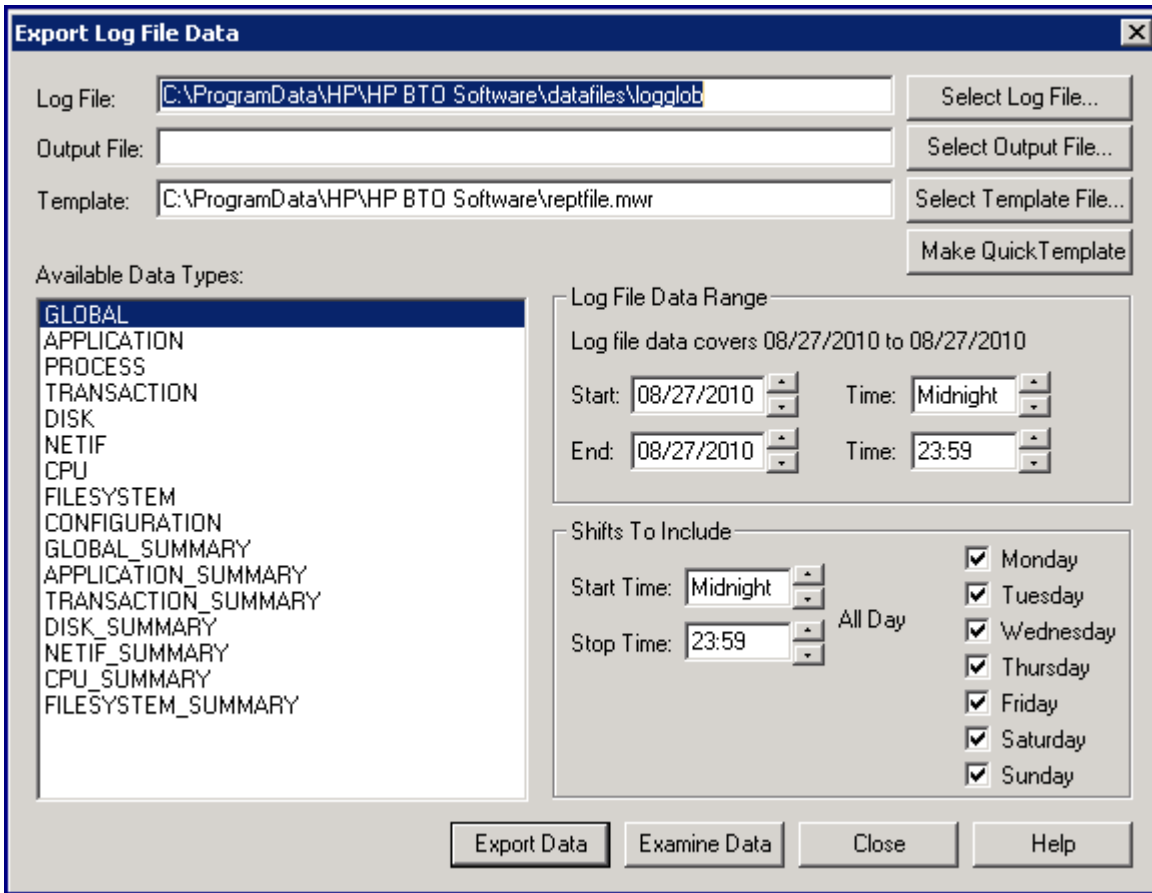
Atributos de archivo

Se pueden asignar varios atributos de archivo a los datos exportados, incluyendo formatos de archivo, valores que representan datos que faltan, separadores de campos, encabezados de columna, número de minutos para intervalos de resumen, diseño, tipos de datos y métricas específicas que se van a incluir en el archivo de exportación. Estos atributos se pueden guardar en los archivos de plantillas de exportación o especificarse directamente utilizando el cuadro de diálogo **Make Quick Template**. (Para obtener más información, consulte [Creación de una plantilla de exportación rápida](#) on page 233.)

Formato de archivos

El formato de salida de un archivo exportado puede ser ASCII, archivo de datos, binario o WK1 (hoja de cálculo):

- El formato ASCII consta de datos de caracteres imprimibles, adecuados para informes impresos o para el post-procesamiento por programas escritos por el usuario o archivos por lotes.



- El formato de archivo de datos es similar al formato ASCII, excepto que todos los elementos no numéricos van entre comillas dobles (" "). El formato de archivo de datos es útil para transferir datos a la mayoría de hojas de cálculo y paquetes gráficos.
- El formato binario no es imprimible. Es más compacto porque los valores numéricos están representados como enteros binarios. Es el formato más adecuado para la entrada en programas de análisis escritos por el usuario, porque es el que necesita menos conversión y mantiene la máxima precisión de las métricas.
- El formato WK1 (hoja de cálculo) es compatible con Microsoft Excel y otros productos de hojas de cálculo, bases de datos y de creación de gráficos.

Valor que falta

Un archivo exportado puede contener el valor de datos que reemplaza a los datos que faltan en el archivo de registro de origen. Los datos que faltan pueden aparecer cuando una métrica no está disponible en determinadas versiones del recopilador `scopent`. Además, los diversos formatos de exportación de diseño para aplicaciones, discos y transacciones reservan espacio en el registro de salida para todas las aplicaciones, discos o transacciones. Si no hay ningún dato registrado para una entrada particular durante un intervalo, sus datos "faltarán".

Separadores de campo

Se puede insertar un carácter de separación de campo entre cada métrica de los archivos exportados de formato ASCII y de archivo de datos. Los caracteres de separación pueden ser imprimibles o no imprimibles (como un carácter de tabulación).

El carácter separador predeterminado es un espacio en blanco, pero algunos programas requieren una coma como separador de campos.

Minutos de resumen

Se puede especificar el número de minutos para cada intervalo de resumen. El número de minutos puede oscilar de cinco a 1.440 minutos (un día).

Encabezados

Se pueden incluir los encabezados de columna en los archivos exportados. El primer registro del archivo son los datos exportados (excepto en los archivos de formato WK1). Sin embargo, si se incluyen encabezados en el archivo, los archivos de formato ASCII y archivo de datos tendrán el título del archivo de exportación (si se especifica) además de encabezados de columna para cada columna de métricas escritos *antes* de los primeros registros de datos del archivo. Los encabezados de los archivos de formato binario se escriben antes del primer registro del archivo y contienen las descripciones de las métricas.

Los archivos WK1 siempre contienen encabezados.

Diseño múltiple

Se pueden especificar diseños múltiples (por salida de registro) para tipos de datos de instancias múltiples, como una aplicación o disco.

El diseño único escribe un registro para cada aplicación o disco que estaba activo durante el intervalo de tiempo. El diseño múltiple escribe un registro para cada intervalo de tiempo, reservando parte de ese registro para cada aplicación o disco configurado.

Título de archivo de exportación

Se puede especificar el título de un archivo de exportación. El título puede contener cadenas literales así como palabras clave de sustitución. Los elementos siguientes se pueden sustituir en la cadena `export title`:

!date	la fecha de creación del archivo de exportación
!time	la hora del día en que se creó el archivo de exportación
!logfile	el nombre del archivo de registro desde el que se obtienen los datos
!collector	el nombre y la versión del programa recopilador (scopent o dsilog)
!class	el tipo de datos solicitados
!system_id	el identificador del sistema que recopila los datos del archivo de registro sin procesar o extraídos de scopent (no es válido con datos de archivo de registro DSI)

Por ejemplo, se puede escribir la cadena siguiente:

```
export "!system_id data from !logfile on !date !time"
```

La cadena puede generar un título similar al siguiente:

```
geminii data from logglob on 10/25/99 08:30 AM
```

Plantillas de archivo de exportación

La tarea de exportación utiliza plantillas de exportación que definen los atributos de archivo para el archivo exportado. Los atributos de archivo predeterminado se toman del archivo <rpmttools>\data\reptfile.mwr que especifica:

- Formato de archivo ASCII
- Un 0 (cero) para el valor que falta
- Un espacio en blanco como separador de campos
- Resúmenes de 60 minutos
- Encabezados incluidos
- En la exportación se incluye un conjunto recomendado de métricas para un tipo de datos o clase determinados

Se puede especificar otro archivo de plantilla de exportación o crear especificaciones de atributos de archivos ad hoc (consulte [Creación de una plantilla de exportación rápida](#) on page 233).

También se pueden crear plantillas de exportación predeterminadas mediante el cuadro de diálogo Configure Export Templates (consulte [Configuración de plantillas de exportación](#) on page 235).

Archivos de exportación predeterminados

Si no se especifica un archivo de salida para que contenga los datos exportados, la tarea de exportación crea un archivo de salida predeterminado en el directorio <unidad_de_disco>:\Archivos de programa\HP\HP BTO Software\data\datafiles basado en el tipo de datos y en el nivel de resumen especificado.

Datos de scopent

Cuando se exportan los datos del archivo de registro `scopent`, se asignan los siguientes nombres de archivo predeterminados a los archivos exportados.

<code>xfrdGLOBAL.ext</code>	datos de detalle globales
<code>xfrsGLOBAL.ext</code>	datos de resumen globales
<code>xfrdAPPLICATION.ext</code>	datos de detalle de aplicación
<code>xfrsAPPLICATION.ext</code>	datos de resumen de aplicación
<code>xfrdPROCESS.ext</code>	datos de detalle de proceso
<code>xfrdDISK.ext</code>	datos de detalle de dispositivo de disco
<code>xfrsDISK.ext</code>	datos de resumen de dispositivo de disco
<code>xfrdCPU.ext</code>	datos de detalle de CPU
<code>xfrsCPU.ext</code>	datos de resumen de CPU
<code>xfrdFILESYSTEM.ext</code>	datos de detalle del sistema de archivos
<code>xfrsFILESYSTEM.ext</code>	datos de resumen del sistema de archivos
<code>xfrdNETIF.ext</code>	datos de detalle de <code>netif</code>
<code>xfrsNETIF.ext</code>	datos de resumen de <code>netif</code>
<code>xfrdTRANSACTION.ext</code>	datos de detalle de seguimiento de transacciones
<code>xfrsTRANSACTION.ext</code>	datos de resumen de seguimiento de transacciones
<code>xfrdCONFIGURATION.ext</code>	datos de detalle de configuración
<code>xfrdLOGICAL.ext</code>	archivo de datos de detalle del sistema lógico
<code>xfrsLOGICAL.ext</code>	archivo de datos de resumen del sistema lógico

Cuando finaliza la tarea de exportación, se puede ver el contenido del archivo de exportación de salida haciendo clic en el botón **Examine Data** del cuadro de diálogo Export Log File Data.

Los nombres de archivo predeterminados se crean a partir del nombre del tipo de datos. El prefijo es `xfrd` o `xfrs` dependiendo de si los datos son datos de detalle o de resumen. La extensión (`.ext`) es el formato de archivo especificado en el archivo de plantilla de exportación: `asc` (ASCII), `bin` (binario), `dat` (archivo de datos) o `wk1` (hoja de cálculo).

Por ejemplo:

`xfrdNETIF.wk1` contiene datos detallados para el tipo de datos NETIF en formato de hoja de cálculo.

`xfrsAPPLICATION.asc` contiene datos de resumen para el tipo de datos de aplicación en formato ASCII.

Datos DSI

Cuando se exportan los datos del archivo de registro DSI, se crean los nombres de archivo predeterminados desde el nombre de la clase. El prefijo es `xfrd` o `xfrs`, dependiendo de si los datos son datos de detalle o de resumen. La extensión es el formato de archivo especificado en el archivo de plantilla de exportación: `asc` (ASCII), `dat` (archivo de datos) o `wk1` (hoja de cálculo).

Por ejemplo:

`xfrdACCTG.wk1` contiene datos detallados para la clase `ACCTG` en formato de hoja de cálculo.

`xfrsPERSONL.asc` contiene datos de resumen para la clase `PERSONL` en formato ASCII.

Para exportar datos de archivos de registro

Para exportar datos de archivos de registro, realice los pasos siguientes:

- 1 Haga clic en **Export** en el menú Logfile de la ventana principal. Aparecerá el cuadro de diálogo Export Log File Data, mostrando los nombres del archivo de registro actualmente activo y el archivo de plantilla de exportación actualmente seleccionado. Se puede especificar un archivo de registro y un archivo de plantilla de exportación diferentes.

Si se especifica un archivo de salida, los datos procedentes de todos los tipos de datos seleccionados se colocan en el mismo archivo.

Si no se especifica un archivo de salida, se crea un archivo de salida predeterminado basado en el tipo de datos o clase y en el nivel de resumen que se especifique. (Consulte [Archivos de exportación predeterminados](#) on page 230)

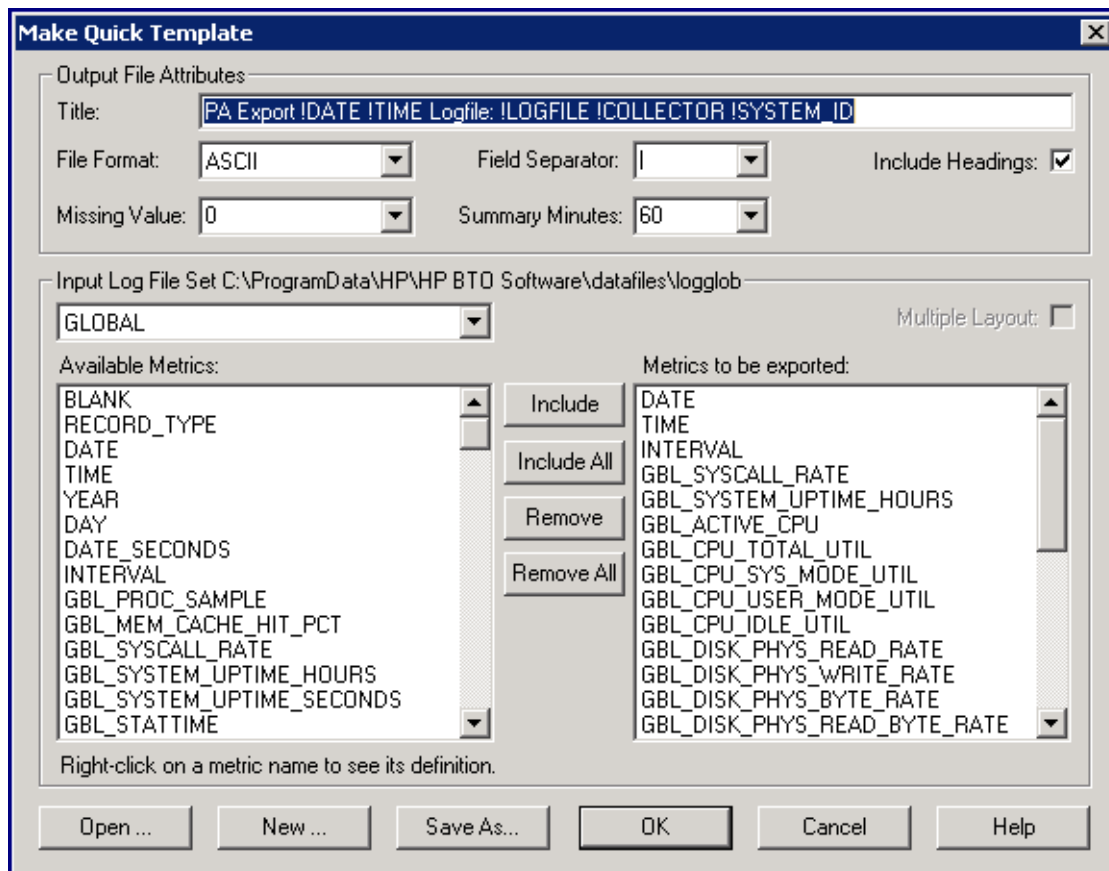
- 2 Después de seleccionar los tipos o clases de datos que se van a exportar, se selecciona el rango de datos del archivo de registro que se van a exportar y los turnos que se van a incluir.
- 3 Para invalidar los atributos de los archivos especificados en la plantilla de exportación predeterminada o para seleccionar métricas específicas que se van a incluir en el archivo de exportación, haga clic en el botón **Make Quick Template**.
- 4 Una vez completados los pasos para crear una plantilla rápida o seleccionadas las métricas específicas para el archivo de exportación, se vuelve al cuadro de diálogo Export Log File. Haga clic en el botón **Export Data** para iniciar el proceso de exportación.
- 5 Cuando finaliza la exportación, se puede ver el contenido del archivo de exportación haciendo clic en el botón **Examine Data**.

Para obtener instrucciones detalladas sobre cómo exportar datos de archivos de registro, elija **Help Topics** en el menú de ayuda, seleccione "**How Do I...?**", y, a continuación, seleccione "**Export log file data**".

Creación de una plantilla de exportación rápida

La función Make Quick Template del cuadro de diálogo Export Log File Data sirve para seleccionar métricas específicas que se van a incluir en el archivo de exportación y para cambiar cualquiera de los atributos y métricas que se especifican en la plantilla de exportación seleccionada para la exportación.

Figura 5 Cuadro de diálogo Make Quick Template



Para crear una plantilla rápida, siga estos pasos:

- 1 Haga clic en el botón **Make Quick Template** en el cuadro de diálogo Export Log File Data. Aparecerá el cuadro de diálogo Make Quick Template, mostrando el título del archivo de exportación seleccionado para la exportación.
- 2 Las casillas que se encuentran debajo de Output File Attributes muestran la configuración actual del archivo de exportación, en función de los atributos de los archivos establecidos en la plantilla de exportación seleccionada para la exportación. Se puede modificar cualquiera de estos ajustes.
- 3 Para usar otro archivo de plantilla de exportación, haga clic en el botón **Open**.
- 4 También se pueden borrar todos los ajustes existentes en el cuadro de diálogo Make Quick Template y crear un archivo de plantilla de exportación totalmente nuevo haciendo clic en el botón **New**.

Selección de métricas para exportación

- Después de seleccionar los tipos o clases de datos de las métricas que se van a exportar, se pueden seleccionar las métricas específicas que se desean incluir en la exportación. Cada tipo o clase de datos cuenta con su propio conjunto de métricas, que se muestran en Available Metrics. Las métricas enumeradas en "Metrics to be exported" son las métricas especificadas por la plantilla de exportación actual para incluir en la exportación. Se puede usar esta lista, suprimir métricas de la lista o seleccionar otras métricas disponibles para incluirlas en la exportación.
- Si se seleccionan los tipos de datos `application`, `disk`, `cpu`, `filesystem`, `netif` o `transaction`, hay que seleccionar la casilla **Multiple Layout** para generar varios diseños (por diseño de registro) o dejarla sin seleccionar para generar un diseño único.

Almacenamiento de las selecciones

Después de haber hecho las selecciones, se puede:

- Continuar con la exportación usando las selecciones realizadas en los atributos de archivo y en la lista de métricas que se van a exportar pero SIN guardar los cambios en ningún archivo de plantilla.
- Guardar las selecciones en un nuevo archivo de plantilla de exportación que se podrá utilizar en futuras exportaciones. El archivo de plantilla de exportación original permanecerá sin cambiar.



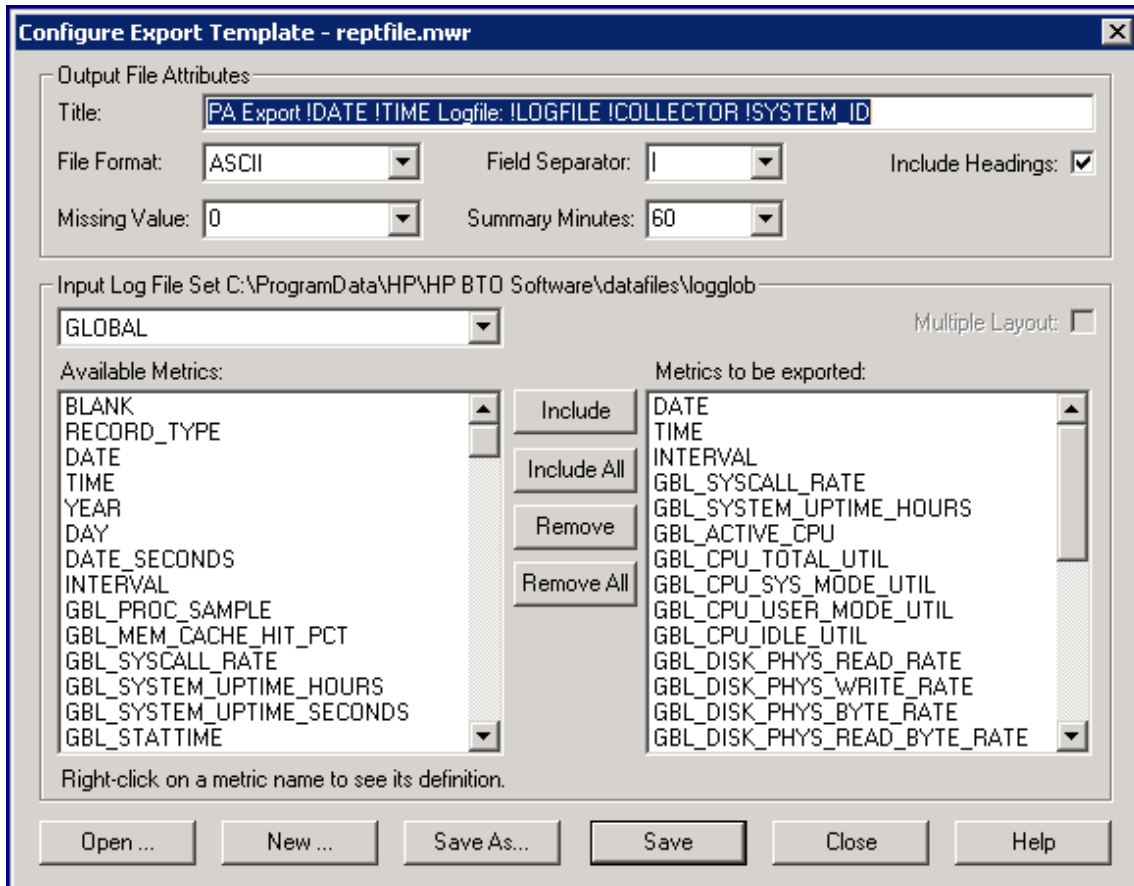
Si se guardan las selecciones en un nuevo archivo de plantilla de exportación, hay que incluir la extensión de nombre de archivo `.mwr` al especificar el nuevo nombre del archivo de plantilla. Por ejemplo, `mytplte.mwr`

Para obtener instrucciones detalladas sobre cómo crear una plantilla de exportación rápida, elija **Help Topics** en el menú de ayuda, seleccione "**How Do I...?**", y, a continuación, seleccione "**Make a quick export template**".

Configuración de plantillas de exportación

El comando **Export Templates** del menú Configure se utiliza para personalizar un archivo de plantilla de exportación existente o para crear uno nuevo. El cuadro de diálogo Configure Export Template se utiliza para seleccionar nuevos atributos de archivos y métricas específicas que se van a incluir en la plantilla.

Figura 6 Cuadro de diálogo Configure Export Template



Para configurar una plantilla de exportación, siga estos pasos:

- 1 Haga clic en **Export Templates** en el menú Configure de la ventana principal. Aparecerá el cuadro de diálogo Configure Export Template mostrando el nombre del archivo de plantilla de exportación abierta en el título del cuadro de diálogo. Para editar otro archivo de plantilla de exportación, haga clic en el botón **Open**.
- 2 Las casillas que se encuentran debajo de Output File Attributes muestran la configuración actual del archivo de exportación, en función de los atributos de los archivos establecidos en la plantilla de exportación que se está configurando. Se puede modificar cualquiera de estos ajustes.
- 3 También se pueden borrar todos los ajustes existentes del cuadro de diálogo Configure Export Template y crear un archivo de plantilla de exportación totalmente nuevo haciendo clic en el botón **New**.

Selección de métricas para exportación

- Después de seleccionar los tipos o clases de datos de las métricas que se van a exportar, se pueden seleccionar las métricas específicas que se desean incluir en la exportación. Cada tipo o clase de datos cuenta con su propio conjunto de métricas, que se muestran en Available Metrics. Las métricas enumeradas en "Metrics to be exported" son las métricas especificadas por la plantilla de exportación actual para incluir en la exportación. Se puede usar esta lista, suprimir métricas de la lista o seleccionar otras métricas disponibles para incluirlas en la exportación.
- Si se seleccionan los tipos de datos `application`, `disk`, `cpu`, `filesystem`, `netif` o `transaction`, hay que seleccionar la casilla **Multiple Layout** para generar varios diseños (por diseño de registro) o dejarla sin seleccionar para generar un diseño único.

Almacenamiento de las selecciones

Existen tres opciones para guardar la plantilla que se ha editado:

- Guardar los cambios en el archivo de plantilla actual.
- Guardar los cambios en un nuevo archivo de plantilla, en cuyo caso el archivo de plantilla original permanece sin modificar.
- Cancelar los cambios para evitar realizar cambios en cualquier archivo de plantilla.



Si se guardan las selecciones en un nuevo archivo de plantilla de exportación, hay que incluir la extensión de nombre de archivo `.mwr` al especificar el nuevo nombre del archivo.

Si se hace clic en el botón **Close** después de realizar los cambios en el archivo de plantilla pero *antes* de guardarlos, aparecerá un mensaje solicitando si se desea cancelar o guardar los cambios. Si se hace clic en el botón **Cancel**, se volverá al cuadro de diálogo Configure Export Template.

Para obtener instrucciones detalladas sobre cómo configurar un archivo de plantilla de exportación, elija **Help Topics** en el menú de ayuda, seleccione "How Do I...?", y a continuación, seleccione "Configure an export template file".

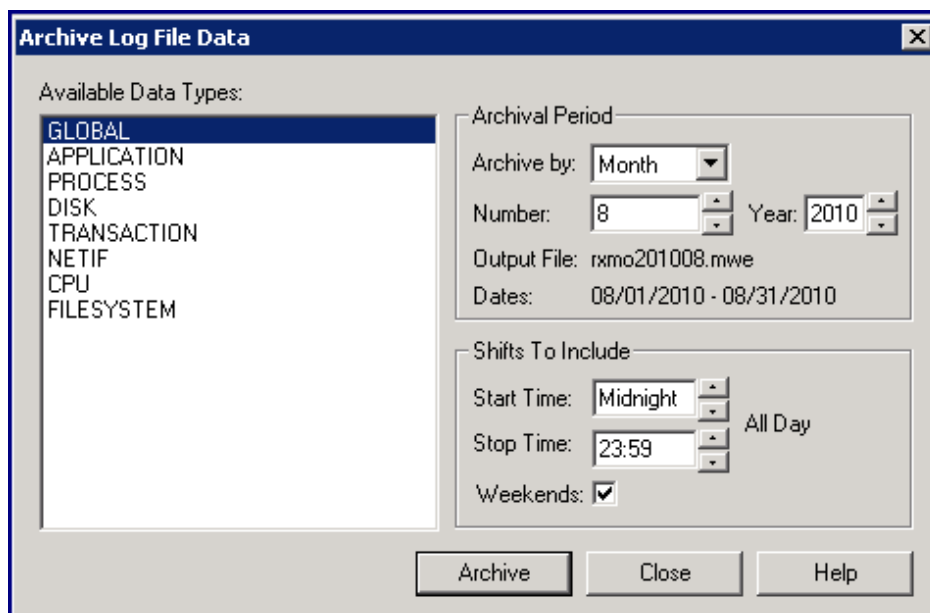
Almacenamiento de datos de archivos de registro

Utilice el comando **Archive** en el menú Logfile para extraer partes seleccionadas de los datos de archivo de registro de `scopent` para su almacenamiento y análisis de datos futuro.

Con fines de almacenamiento, los datos sólo se pueden extraer de los archivos de registro sin procesar.

Los datos extraídos se colocan automáticamente en un archivo de salida de almacenamiento en el directorio `<unidad_de_disco>:\Archivos de programa\HP\HP BTO Software\data\datafiles`, cuyo nombre refleja el período de almacenamiento seleccionado. Estos archivos se pueden copiar en una cinta para su almacenamiento sin conexión y después eliminarse para liberar espacio en disco.

Figura 7 Cuadro de diálogo Archive Log File Data



Períodos de almacenamiento

Los datos se pueden seleccionar para su extracción en función de los datos registrados durante un período semanal, mensual o anual específico.

También se puede limitar la extracción de datos registrados durante determinadas horas del día que se correspondan a turnos de trabajo o excluir fines de semana (sábado y domingo). Si no se especifica ningún turno, el valor predeterminado es un valor de 24 horas para la extracción de datos para cada día. De manera predeterminada, se incluyen los fines de semana.

Anexión de datos almacenados

La función de almacenamiento tiene una característica especial. Dependiendo del período de almacenamiento que se haya seleccionado (semanal, mensual o anual), el archivo de salida *anterior* de dicho período

se comprueba automáticamente para ver si contiene datos extraídos hasta el último día. Si no es así, los datos se anexan al archivo para completar la extracción del período de almacenamiento anterior.

Por ejemplo, el 7 de mayo de 1999 se comienzan a almacenar los datos mensuales para mayo de 1999. Se crea un archivo de salida llamado `rxmo199905.mwe`, que contiene los datos desde el 1 de mayo hasta la fecha actual (7 de mayo).

El 4 de junio de 1999, se llama a otro período de almacenamiento mensual. Antes de que se cree el archivo `rxmo199906.mwe` para junio, se comprueba el archivo `rxmo199905.mwe` del mes anterior. Cuando se encuentra que está incompleto, se anexa a él los datos para completar la extracción hasta el 31 de mayo de 1999. A continuación, se crea el archivo `rxmo199906.mwe` para que almacene los datos desde el 1 de junio de 1999 hasta la fecha actual (4 de junio).

Mientras se invoque otro período de almacenamiento mensual (semanal, anual) al menos una vez al mes (semana, año), esta característica completará cada archivo del período de almacenamiento antes de crear el siguiente archivo del período de almacenamiento.

Sugerencias de almacenamiento

A continuación se indican algunas sugerencias para almacenar los datos del archivo de registro:

- Una vez al mes, hay que especificar el período de almacenamiento mensual y extraer todos los datos de detalle de los archivos de registro sin procesar en un único archivo de registro extraído.
- Si el sistema genera más de 64 MB de datos cada mes, es posible que haya que extraer datos semanalmente, o bien eliminar los datos de detalle del proceso en la extracción.
- Hay que extraer los datos de resumen de la aplicación y los resúmenes globales anualmente, lo que debería minimizar el espacio en disco requerido. Este archivo de almacenamiento se puede utilizar entonces para el análisis de tendencias a largo plazo.

Para almacenar datos de archivos de registro, realice los pasos siguientes:

- 1 Haga clic en **Archive** en el menú Logfile de la ventana principal. Se mostrará el cuadro de diálogo Archive Log File Data. Los datos que se van a almacenar se extraen del conjunto de archivos de registro sin procesar.
- 2 Después de seleccionar el tipo de datos que se van a almacenar en la lista Available Data Types, hay que especificar el período de almacenamiento (semana, mes o año) y los turnos que haya que incluir.
- 3 Haga clic en el botón **Archive** para iniciar el proceso de almacenamiento.

Para obtener instrucciones detalladas sobre cómo almacenar datos de archivos de registro, elija **Help Topics** en el menú de ayuda, seleccione "**How Do I...?**", y, a continuación, seleccione "**Archive log file data**".

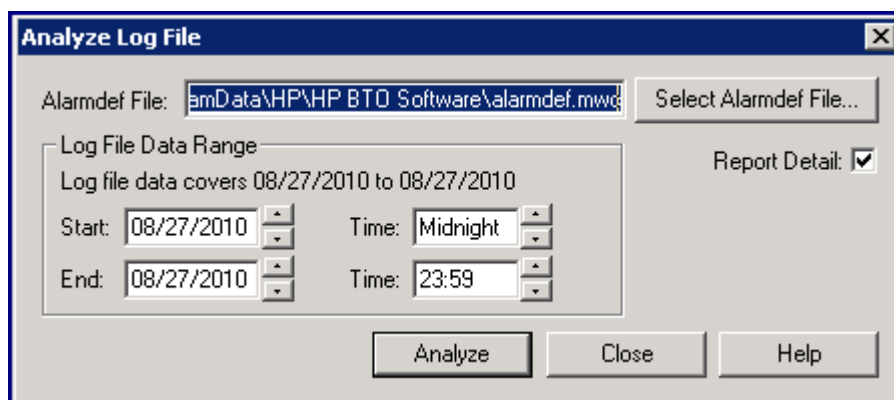
Análisis de un archivo de registro

Se utiliza el comando **Analyze Log File** del menú Logfile para analizar datos en el conjunto de archivos de registro sin procesar, comparándolos con las definiciones de alarma en el archivo de definiciones de alarma, e informar de cualquier actividad de alarma resultante.

Esta tarea permite evaluar si las definiciones de alarma se corresponden o no con los datos históricos recopilados en el sistema. Permite además al usuario decidir si sus definiciones de alarma generarán muchas o pocas alarmas en el sistema de análisis.

Se hace referencia a los archivos de registro sin procesar analizados en SCOPE, el origen de datos predeterminado de Componente Performance Collection. Para analizar un archivo de registro diferente, hay que colocar una instrucción USE en el archivo de definiciones de alarma que especifique el nombre del origen de datos que haga referencia a ese archivo de registro.

Figura 8 Cuadro de diálogo Analyze Log File



Rango de datos que se van a analizar

Se pueden analizar los datos del archivo de registro que se recopilaron durante un período específico de tiempo. Si no se especifica un rango concreto de datos que se van a analizar, los datos se analizan usando la fecha de inicio predeterminada, es decir, 30 días después de la última fecha del archivo de registro o, si hay menos de 30 días de datos, la fecha del primer registro del archivo de registro.

Informe de análisis

Cuando se ejecuta esta tarea, se genera un informe imprimible que enumera los eventos de alarma y un resumen de alarmas. (Los eventos de alarma sólo se enumeran si está seleccionada la casilla Report Detail del cuadro de diálogo Analyze Log File.)

- Los eventos de alarma incluyen el estado START, END y REPEAT de la alarma, más cualquier texto de instrucciones PRINT asociadas. Además, si se enumera cualquier texto en las instrucciones PRINT como condiciones (en instrucciones IF) y se cumplen, se incluye el texto. Las instrucciones EXEC no se ejecutarán pero sí se mostrarán de manera que el usuario pueda ver lo que se hubiera ejecutado.

- Los resúmenes de alarmas muestran un recuento del número de alarmas que han tenido lugar y del tiempo total en que cada alarma ha estado activa (on). El recuento incluye los inicios y repeticiones de alarmas, pero no la finalización de las alarmas.

Para analizar un archivo de registro, siga estos pasos:

- 1 Haga clic en **Analyze** en el menú Logfile de la ventana principal. Aparecerá el cuadro de diálogo **Analyze Log File** mostrando el nombre del archivo de definiciones de alarma actualmente seleccionado.
- 2 Para usar otro archivo de definiciones de alarma, haga clic en el botón **Select Alarmdef File**.
- 3 Seleccione el rango de datos de archivo de registro que se van a analizar.
- 4 Seleccione la casilla **Report Detail** si se desean incluir eventos de alarma en el informe de análisis. En caso contrario, sólo se genera el resumen de alarmas.
- 5 Haga clic en el botón **Analyze** para iniciar el análisis. Los resultados del análisis se muestran en una ventana Report Viewer de MeasureWare Agent.

Para obtener instrucciones detalladas sobre cómo analizar un archivo de registro, elija **Help Topics** en el menú de ayuda, seleccione "How Do I...?", y, a continuación, seleccione "**Analyze a log file**".

Exploración de un archivo de registro

El comando **Scan Log File** del menú Logfile se utiliza para explorar un archivo de registro `scopent` y crear un informe sobre su contenido. O bien se puede explorar un archivo de registro completo o partes de un archivo de registro en busca de datos recopilados durante un período específico de tiempo.

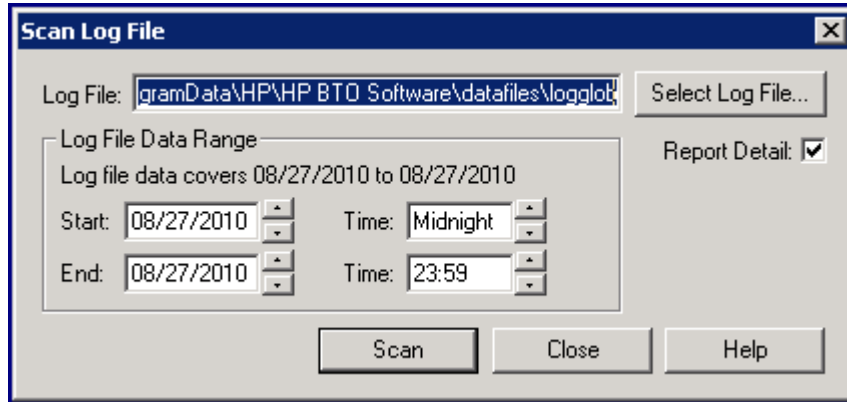
El informe producido por la exploración consta de 12 secciones. Las cuatro secciones siguientes del informe siempre se imprimen.

- Informe de resumen de proceso
- Resumen de cobertura del recopilador
- Resumen de contenidos del archivo de registro
- Resumen de espacio libre del archivo de registro

Las siguientes ocho secciones del informe sólo se imprimen si se selecciona **Report Detail** en el cuadro de diálogo Scan Log File.

- Información global inicial del archivo `parm` e información de configuración del sistema
- Definiciones de aplicación iniciales del archivo `parm`
- Cambios globales del archivo `parm`
- Notificaciones de cambios/aplicación del archivo `parm`
- Notificaciones de tiempo de inactividad del recopilador
- Informes de resúmenes específicos de aplicación

Figura 9 Cuadro de diálogo Scan Log File



Para explorar un archivo de registro, siga estos pasos:

- 1 Haga clic en **Scan Log File** en el menú Logfile de la ventana principal. Aparecerá el cuadro de diálogo Scan Log File con el nombre del archivo de registro actualmente seleccionado resaltado.
- 2 Para explorar otro archivo de registro, haga clic en el botón **Select Log File**.
- 3 Para explorar datos que se registraron durante un período de tiempo específico, en Log File Data Range, seleccione o escriba las fechas y horas para el comienzo y el fin de dicho período de tiempo.
- 4 Seleccione la casilla **Report Detail** si se desea un informe de exploración completo. En caso contrario, sólo se generará un subconjunto del informe.
- 5 Para iniciar el proceso de exploración, haga clic en el botón **Scan**. Los resultados de la exploración se muestran en una ventana Report Viewer de Componente Performance Collection.

Para obtener instrucciones detalladas sobre cómo explorar un archivo de registro, elija **Help Topics** en el menú de ayuda, seleccione "How Do I...?", y a continuación, seleccione "**Scan a log file**".

Cambio de tamaño de un archivo de registro

El comando **Resize Log File** en el menú Logfile se utiliza para cambiar el tamaño de los archivos de registro sin procesar de scopent. Se puede cambiar el tamaño utilizando el tamaño en megabytes del archivo especificado o el número de días de datos que debe almacenar el archivo.

El tamaño máximo de un archivo de registro sin procesar se especifica en el parámetro size del archivo parm. El cambio de tamaño del archivo de registro ofrece más control sobre la frecuencia de reversión de los datos del archivo de registro.

Se puede seleccionar cualquiera de los tipos de datos siguientes para cambiar el tamaño: global, application, process, device o transaction, que se corresponden con los archivos de registro sin procesar logglob, logappl, logproc, logdev y logtran. Después, se selecciona cómo se va a realizar el cambio de tamaño, en megabytes o por número de días. Dependiendo del tipo de cambio de tamaño que se elija, el cuadro Log File Settings del cuadro de diálogo Resize Log File muestra lo siguiente:

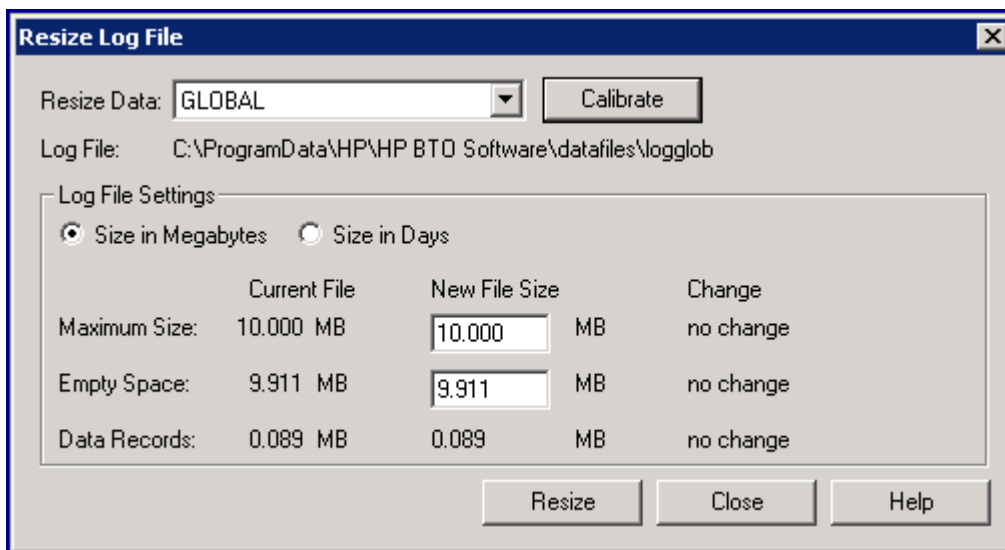
- Los campos Maximum Size muestran el tamaño del archivo actual, el nuevo tamaño del archivo y la modificación realizada por el cambio de tamaño.
- Los campos Empty Space muestran la cantidad de espacio del archivo actual, la cantidad requerida en el archivo después de completado el proceso de cambio de tamaño y el cambio. Estos valores se usan para determinar si alguno de los datos que se encuentra en el archivo de registro deben eliminarse en el proceso de cambio de tamaño.
- Los campos Data Records muestran la cantidad de registros de datos contenidos en el archivo de registro actual y la nueva cantidad de registros de datos que se encontrarán en el archivo de registro cuyo tamaño se ha cambiado.

Los tamaños del archivo de registro se mantienen en megabytes. Con frecuencia es más conveniente especificar los tamaños en días en lugar de en megabytes. Si se selecciona "Size in Days", todas las unidades del cuadro de diálogo cambiarán a "días". La conversión de megabytes a días se basa en un valor de "megabytes-por-día" para cada tipo de datos. Inicialmente, se utilizan valores estimados para esta conversión.

Se puede obtener un valor más preciso haciendo clic en el botón **Calibrate**.

La función de calibración mide en realidad los archivos de registro existentes en busca de valores megabytes-por-día más precisos. Si se está especificando el tamaño en megabytes, no es necesaria ninguna conversión y no hay que utilizar la función de calibración.

Figura 10 Cuadro de diálogo Resize Log File



Antes de cambiar el tamaño de un archivo de registro, se *debe* detener el recopilador `scopent`. Para detener `scopent`, siga los pasos de [Detención y reinicio de la recopilación de datos](#) on page 40.

Intentar cambiar el tamaño de un archivo de registro sin detener primero `scopent` no afectará al archivo de registro existente. Para cambiar el tamaño de un archivo de registro, siga estos pasos:

- 1 Una vez detenido `scopent`, elija **Resize Log File** en el menú Logfile de la ventana principal para mostrar el cuadro de diálogo Resize Log File.
- 2 En el cuadro Resize Data, seleccione el tipo de datos cuyo tamaño se va a cambiar: **global**, **application**, **process**, **device** o **transaction**.
- 3 Seleccione **Size in Megabytes** o **Size in Days**. Dependiendo de lo que se haya seleccionado, se muestran Current File y New File Sizes.

- 4 Para realizar el cambio de tamaño basado en la opción **New File Sizes**, haga clic en el botón **Resize** para iniciar el proceso de cambio de tamaño.

Para obtener una estimación más precisa de la cantidad de espacio adicional que hay que agregar al archivo de registro cuando se cambia el tamaño en días, siga estos pasos:

- 1 Haga clic en el botón **Calibrate**. Poco después, se mostrará el número y el tamaño real de los registros de datos que se registraron en el archivo durante los últimos 30 días.
- 2 Haga clic en el botón **Close** para volver al cuadro de diálogo **Resize Log File**. Se mostrarán los valores de **New Current File** y **New File Size** basados en la calibración, si se cambia el tamaño en días.
- 3 Haga clic en el botón **Resize** para cambiar el tamaño del archivo de registro.
- 4 Antes de realizar otra tarea, inicie **scopent** mediante los pasos de [Detención y reinicio de la recopilación de datos](#) on page 40.

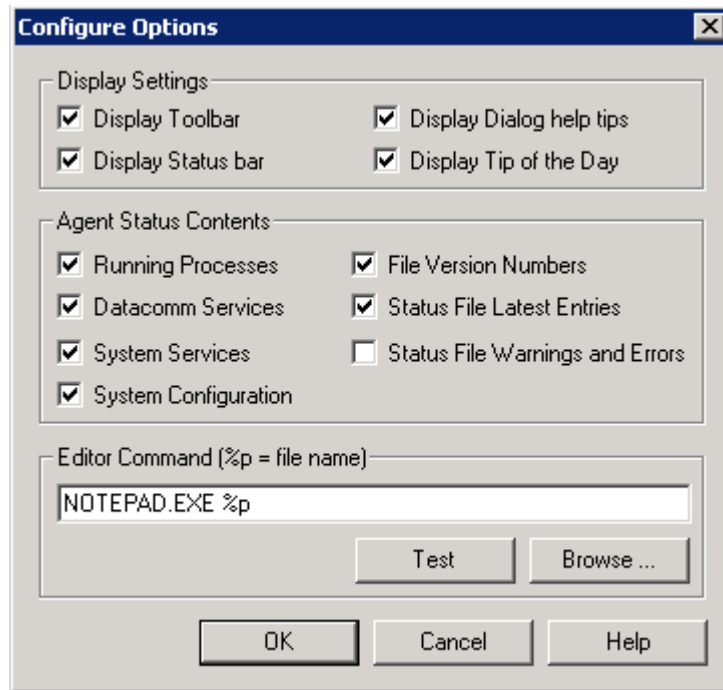
Para obtener instrucciones detalladas sobre cómo cambiar el tamaño de un archivo de registro, elija **Help Topics** en el menú de ayuda, seleccione "**How Do I...?**," y, a continuación, seleccione "**Resize a log file**".

Configuración de opciones de usuario

El comando **Options** del menú **Configure** se utiliza para controlar la visualización de la barra de herramientas, barra de estado, sugerencias de ayuda de cuadros de diálogo y de la sugerencia del día en la ventana principal y mientras se está usando **Componente Performance Collection**.

El comando **Options** también se utiliza para configurar un editor o procesador de texto para modificar los parámetros de recopilación y archivos de definiciones de alarma y para seleccionar el tipo de información de estado que se desea ver cuando se elige el comando **Status** en el menú **Agent**.

Figura 11 Cuadro de diálogo Configure Options



Para configurar las opciones de usuario, siga estos pasos:

- 1 Haga clic en el comando **Options** en el menú Configure de la ventana principal para mostrar el cuadro de diálogo Configure Options.
- 2 Seleccione la casilla **Display Toolbar** para mostrar la barra de herramientas de la ventana principal.
- 3 Seleccione la casilla **Display Status Bar** para mostrar el estado actual en la parte inferior de cada cuadro de diálogo y en la ventana principal.
- 4 Seleccione la casilla **Display Dialog Help Tips** para mostrar las sugerencias de ayuda dentro de los cuadros de diálogo.
- 5 Seleccione la casilla Display Tip of the Day para mostrar la sugerencia del día cuando abra la ventana principal de Componente Performance Collection.

Para realizar la configuración, siga estos pasos:

- 1 Escriba el nombre de archivo y la ruta del directorio del editor en el cuadro Editor Command, utilizando la extensión de nombre de archivo .exe (por ejemplo, C:\MSOffice\winword\winword.exe).
- 2 Haga clic en el botón **Browse** para mostrar el cuadro de diálogo Select a Text Editor en el que puede seleccionar el editor.
- 3 Haga clic en el botón **Test** para asegurarse de que el editor seleccionado está configurado y, a continuación, haga clic en **OK**.

Para configurar la información de estado del agente que desea ver, seleccione una o varias de las casillas de opción que aparecen en Agent Status Contents y, a continuación, haga clic en **OK**.

Para obtener instrucciones detalladas sobre cómo configurar las opciones de usuario, elija **Help Topics** en el menú de ayuda, seleccione "How Do I...?", y, a continuación, seleccione "Configure user options".

Configuración de los parámetros de recopilación

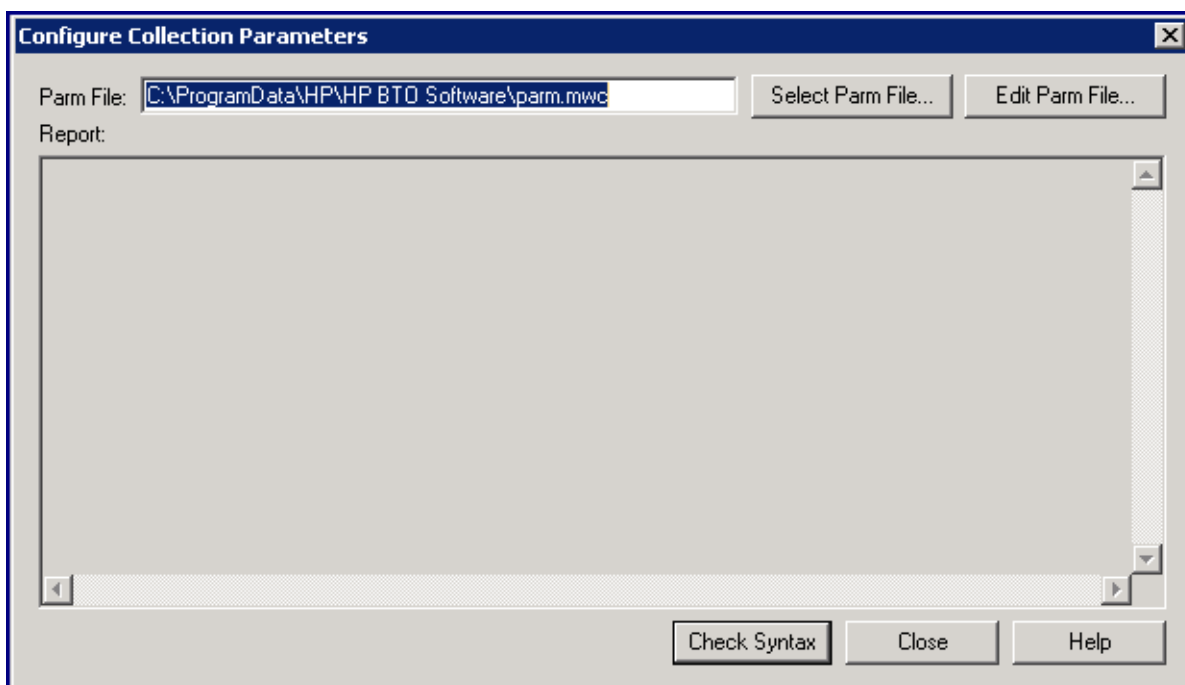
El comando **Collection Parameters** del menú **Configure** se utiliza para comprobar la sintaxis del archivo `parm` que utiliza `scopent` para la recopilación de datos. Se puede examinar la configuración del archivo

`parm` para buscar errores de sintaxis y advertencias y ver cuánto espacio hay disponible para definir las aplicaciones.

Si se encuentran errores o advertencias y se desea corregirlos, o si se desea modificar o agregar parámetros al archivo `parm`, se puede modificar fácilmente el archivo `parm` mediante la función **Edit Parm File**.

En [Administración de la recopilación de datos](#) on page 15 se puede encontrar una descripción detallada del archivo `parm` y sus parámetros.

Figura 12 Cuadro de diálogo **Configure Collection Parameters**



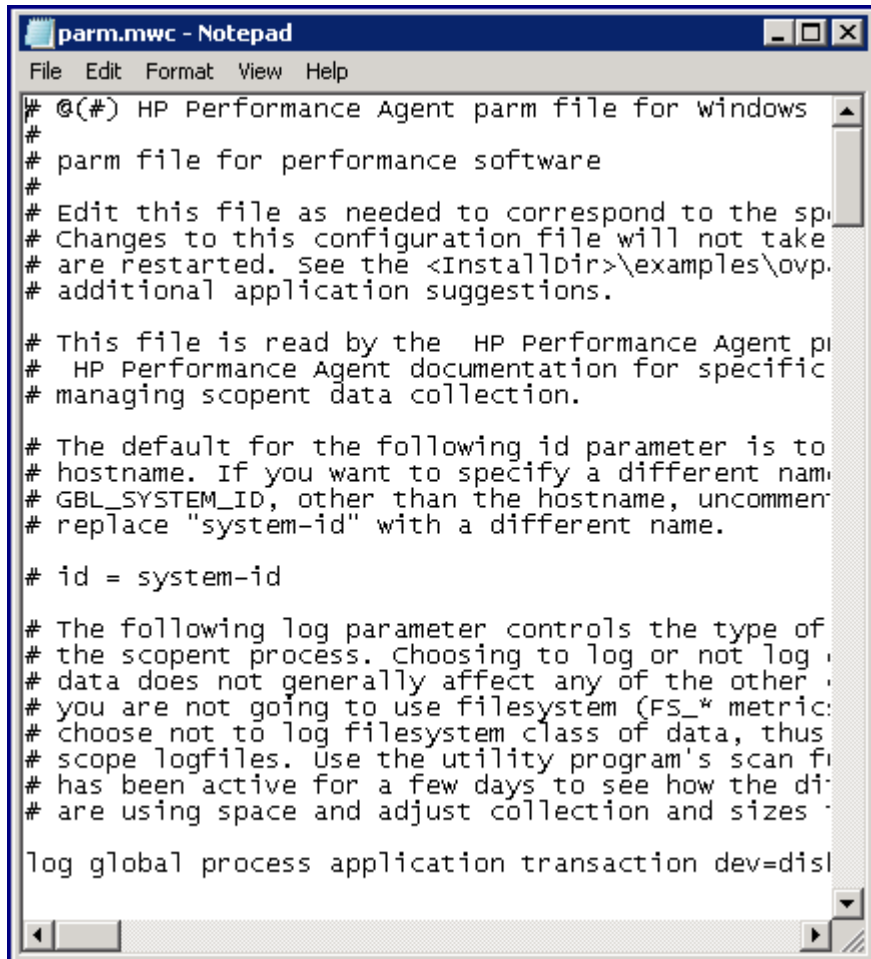
Para comprobar la sintaxis, siga estos pasos:

- 1 Haga clic en **Collection Parameters** en el menú **Configuration** de la ventana principal de **Componente Performance Collection**. Aparecerá el cuadro de diálogo **Configure Collection Parameters** mostrando el nombre del archivo `parm.mwc` actualmente abierto en el cuadro **Parm File**.
- 2 Para comprobar otro archivo `parm`, haga clic en el botón **Select Parm File**.
- 3 Para comprobar la sintaxis del archivo `parm`, haga clic en el botón **Check Syntax**. Los errores o advertencias resultantes se mostrarán en la ventana **Report Viewer** de **Componente Performance Collection**.
- 4 Para modificar cualquier parte del archivo `parm`, haga clic en el botón **Edit Parm File**. Puede colocar los cuadros de diálogo **Edit Parm File** y **Configure Collection Parameters** en la pantalla para poder utilizar ambos al mismo tiempo.

Para obtener instrucciones detalladas sobre cómo comprobar la sintaxis del archivo `parm`, elija **Help Topics** en el menú de ayuda, seleccione "**How Do I...?**", y, a continuación, seleccione "**Check the syntax of a collection parameters file**".

Modificación de un archivo de parámetros de recopilación

Figura 13 Ventana **Modify Collection Parameters File**



```
parm.mwc - Notepad
File Edit Format View Help
# @(#) HP Performance Agent parm file for windows
#
# parm file for performance software
#
# Edit this file as needed to correspond to the sp
# Changes to this configuration file will not take
# are restarted. see the <InstallDir>\examples\ovp
# additional application suggestions.
#
# This file is read by the HP Performance Agent pi
# HP Performance Agent documentation for specific
# managing scopent data collection.
#
# The default for the following id parameter is to
# hostname. If you want to specify a different nam
# GBL_SYSTEM_ID, other than the hostname, uncommen
# replace "system-id" with a different name.
#
# id = system-id
#
# The following log parameter controls the type of
# the scopent process. Choosing to log or not log
# data does not generally affect any of the other
# you are not going to use filesystem (FS_* metric:
# choose not to log filesystem class of data, thus
# scope logfiles. Use the utility program's scan fi
# has been active for a few days to see how the di
# are using space and adjust collection and sizes
#
log global process application transaction dev=disl
```

Para modificar el archivo `parm`, siga estos pasos:

- 1 Haga clic en **Collection Parameters** en el menú **Configuration** de la ventana principal de **Componente Performance Collection** y, a continuación, haga clic en el botón **Edit Parm File** en el cuadro de diálogo **Configure Collection Parameters**. Se mostrará el contenido del archivo `parm` actualmente abierto en el editor o procesador de texto previamente especificado.
(Para especificar un editor o procesador de texto, consulte [Configuración de opciones de usuario](#) on page 243.)
- 2 Antes de realizar los cambios en el archivo, consulte [Archivo parm](#) on page 18 para obtener información sobre algunas reglas y convenciones que se deben seguir.

3 Modifique el archivo, si es preciso, y guárdelo en formato de texto.

- ▶ No se puede modificar el archivo `parm` con el editor **WordPad** de Windows mientras Componente Performance Collection se esté ejecutando. Hay que detener Componente Performance Collection, utilizar **WordPad** y, a continuación, reiniciar Componente Performance Collection. Sin embargo, se puede usar el editor **Notepad** para modificar el archivo mientras Componente Performance Collection se está ejecutando.

Antes de continuar con otra tarea, *hay* que activar los cambios realizados en el archivo `parm`. Siga estos pasos:

- 1 Haga clic en **Start/Stop** en el menú Agent de la ventana principal de Componente Performance Collection para abrir la ventana MeasureWare Services.
- 2 Seleccione la casilla **Data Collection**.
- 3 Haga clic en el botón **Refresh**.
- 4 Haga clic en el botón **Close** para volver a la ventana principal.

Para obtener instrucciones detalladas sobre cómo modificar el archivo `parm`, elija **Help Topics** en el menú de ayuda, seleccione "**How Do I...?**," y, a continuación, seleccione "**Modify a collection parameters file**".

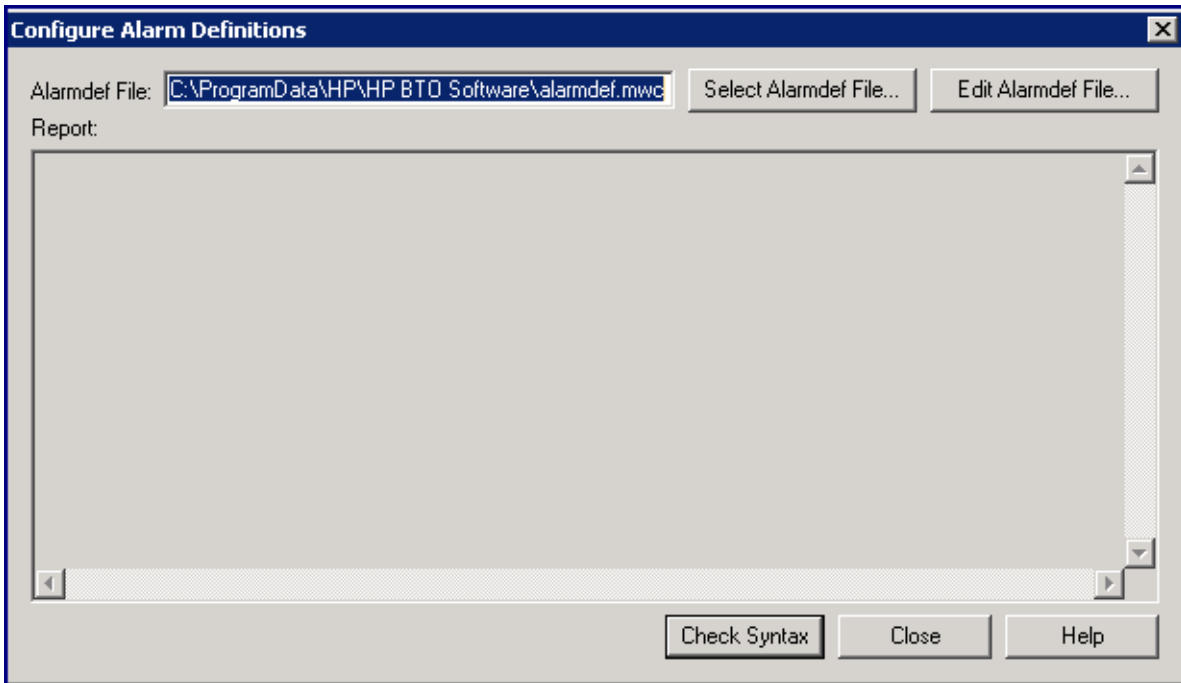
- ▶ Si se utiliza WordPad, Notepad o Microsoft Word para modificar el archivo `parm.mwc` y después se usa el comando **Save As** para guardarlo, se agregará automáticamente la extensión `.txt` predeterminada al nombre del archivo. Se dispondrá entonces de un archivo denominado `parm.mwc.txt`. Para conservar el nombre de archivo `parm.mwc`, se utiliza el comando **Save As** para guardar el archivo como un archivo de texto y se rodea el nombre de archivo entre comillas ("). Por ejemplo: "`parm.mwc`".

Configuración de definiciones de alarma

Se puede utilizar el comando **Alarm Definitions** en el menú Configure para comprobar la sintaxis de las definiciones de alarma en un archivo de definiciones de alarma (`alarmdef.mwc`). Una vez determinada que la sintaxis de las definiciones de alarma es correcta, se puede analizar un archivo de registro comparándolo con las definiciones de alarma para buscar alarmas en el archivo histórico de registro (consulte [Análisis de un archivo de registro](#) on page 239).

Si se encuentran errores o advertencias y se desea corregirlos, o si se desea agregar o eliminar definiciones de alarma, se puede modificar el archivo de definiciones de alarma fácilmente con el botón **Edit Alarmdef File** en el cuadro de diálogo Configure Alarm Definitions.

Figura 14 Cuadro de diálogo Configure Alarms Definitions



Para comprobar la sintaxis, siga estos pasos:

- 1 Haga clic en **Alarms Definitions** en el menú Configure de la ventana principal de Componente Performance Collection. Aparecerá el cuadro de diálogo Configure Alarm Definitions mostrando el nombre del archivo de definiciones de alarma actualmente abierto.
- 2 Para comprobar otro archivo de definiciones de alarma, haga clic en el botón **Select Alarmdef File**.
- 3 Haga clic en el botón **Check Syntax** para iniciar el proceso de comprobación de sintaxis. Tras unos pocos segundos, se muestran los resultados de la comprobación, incluyendo errores o advertencias, en la ventana Report Viewer de Componente Performance Collection.
- 4 Para modificar cualquier parte del archivo de definiciones de alarma, haga clic en el botón **Edit Alarmdef File**.

Para obtener instrucciones detalladas sobre cómo comprobar la sintaxis de un archivo de definiciones de alarma, elija **Help Topics** en el menú de ayuda, seleccione "**How Do I...?**", y, a continuación, seleccione "**Checking the syntax of an alarm definitions file**".

Modificación de un archivo de definiciones de alarma

Para modificar el archivo alarmdef, siga estos pasos:

- 1 Haga clic en **Alarm Definitions** en el menú Configure de la ventana principal de Componente Performance Collection y, a continuación, haga clic en el botón **Edit Alarmdef File** en el cuadro de diálogo Configure Alarm Definitions. Se mostrará el contenido del archivo de definiciones de alarma actualmente abierto en el editor o procesador de texto previamente especificado. (Para configurar un editor o procesador de texto, consulte [Configuración de opciones de usuario](#) on page 243.)
- 2 Antes de realizar los cambios en el archivo, consulte [Referencia de sintaxis de alarma](#) on page 166 para obtener información sobre definiciones de alarma.

- 3 Modifique el archivo, si es preciso, y guárdelo en formato de texto.



Si se utiliza WordPad, Notepad o Microsoft Word para modificar el archivo de definiciones de alarma y después se usa el comando **Save As** para guardarlo, se agregará automáticamente la extensión `.txt` predeterminada al nombre del archivo. Se dispondrá entonces de un archivo denominado `alarmdef.mwc.txt`. Para conservar el nombre de archivo `alarmdef.mwc.txt`, se utiliza el comando **Save As** para guardar el archivo como un archivo de texto y se rodea el nombre de archivo entre comillas ("). Por ejemplo, "alarmdef.mwc".

Activación de cambios

Antes de continuar con otra tarea, *hay* que activar los cambios realizados en el archivo de definiciones de alarma. Siga estos pasos:

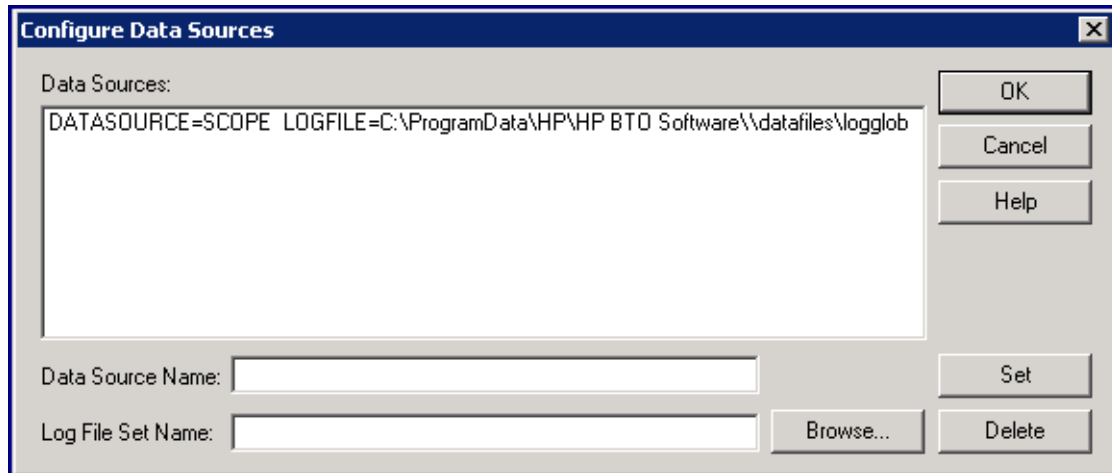
- 1 Haga clic en **Start/Stop** en el menú Agent de la ventana principal de Componente Performance Collection para abrir la ventana MeasureWare Services.
- 2 Seleccione la casilla **Alarm Definitions**.
- 3 Haga clic en el botón **Refresh**.
- 4 Haga clic en el botón **Close** para volver a la ventana principal.

Para obtener instrucciones detalladas sobre cómo modificar un archivo de definiciones de alarma, elija **Help Topics** en el menú de ayuda, seleccione "How Do I...?", y, a continuación, seleccione "**Modify an alarm definitions file**".

Configuración de orígenes de datos

Performance Collection Component utiliza orígenes de datos para cada origen de datos específico, como archivos de registro `scopent` o archivos de registro DSI. Cada origen de datos consta de un único conjunto de archivos de registro. El origen de datos está configurado en el archivo `datasources`, que reside en el directorio `<DataDir>\conf\perf`. Cuando se inicia por primera vez Componente Performance Collection después de la instalación, un origen de datos predeterminado llamado SCOPE ya está configurado y proporciona un conjunto de archivos de registro `scopent`.

Figura 15 Cuadro de diálogo Configure Data Sources



Formato de los archivos de orígenes de datos

Cada entrada que se introduce en el archivo `datasources` representa un origen de datos que consta de un único conjunto de archivos de registro.

La entrada especifica el nombre del origen de datos por el cual se conoce el servidor de repositorio y dónde se van a encontrar los datos que contiene.

Las entradas no distinguen entre mayúsculas y minúsculas. La sintaxis es la siguiente:

datasource=nombre_de_origen_de_datos logfile=conjunto_archivos_de_registro

- **datasource** es una palabra clave. *nombre_de_origen_de_datos* es el nombre utilizado para identificar el origen de datos usado en las definiciones de alarma o en el análisis de software. Los nombres de los orígenes de datos deben ser únicos. La longitud máxima de *nombre_de_origen_de_datos* es de 64 caracteres.
- **logfile** es una palabra clave. *conjunto_archivos_de_registro* es el nombre completo que identifica el conjunto de archivos de registro. Puede ser un conjunto de archivos de registro sin procesar creado por `scopent`, un archivo de registro extraído creado por la tarea `extract` o un conjunto de archivos de registro DSI. Si se especifica un nombre de ruta del archivo de registro que contenga espacios en blanco incrustados, hay que rodear el nombre de ruta con comillas (").

Al especificar un conjunto de archivos de registro `scopent`, sólo se utiliza el nombre de archivo `logglob`. No hay que especificar otros nombres de archivo de registro sin procesar porque a ellos se accede como un conjunto de archivos de registro único.

Lo mismo se aplica al especificar un conjunto de archivos de registro DSI.

Sólo hay que especificar el nombre del archivo raíz DSI. No hay que especificar ninguno de los otros archivos del conjunto de archivos de registro DSI.

Configuración de orígenes de datos desde ubicaciones remotas

Se requiere la convención de nomenclatura universal (UNC) cuando se especifica un conjunto de archivos de registro que reside en un recurso compartido en red. Al iniciar el sistema, el servicio de Componente Performance Collection se inicia automáticamente y las asignaciones de unidades para sistemas de archivos conectados de forma remota no se establecen hasta que el usuario inicia sesión. Por tanto, cualquier origen de datos que utilice un nombre asignado a una unidad para hacer referencia a un archivo de registro en un sistema remoto hace que Coda genere un error de origen de datos no válido. Si se inicia el servicio de Componente Performance Collection *después* del inicio de sesión, el origen de datos se procesa debido al establecimiento de las asignaciones de unidad.

A continuación se muestran tres ejemplos de entradas de orígenes de datos:

Ejemplo 1:

En el ejemplo siguiente se muestra el origen de datos SCOPE predeterminado que reside en el directorio

```
<unidad_de_disco>:\Archivos de programa\HP\HP BTO  
Software\data\datafiles\predeterminado.
```

```
datasource=SCOPE logfile="C:\Archivos de programa\HP\HP BTO  
Software\data\datafiles\logglob"
```

Ejemplo 2:

En el ejemplo siguiente se utiliza la convención de nomenclatura universal (UNC) para especificar un conjunto de archivos de registro que reside en un recurso compartido en red.

```
datasource=RXLOG logfile=\\lab_sys\my_share\rxlog
```

Ejemplo 3:

En el ejemplo siguiente se muestra el origen de datos SCOPE que reside en un directorio cuyo nombre de ruta contiene un espacio en blanco incrustado.

```
datasource=SCOPE logfile="C:\Archivos de programa\HP\HP BTO Software\  
data\donna test\logglob"
```

Para configurar orígenes de datos, haga clic en **Data Sources** en el menú Configure de la ventana principal de Componente Performance Collection.

Se mostrará el cuadro de diálogo Configure Data Sources mostrando las entradas de orígenes de datos actuales. Cada entrada representa un único origen de datos.

Para realizar la modificación, siga estos pasos:

- 1 Seleccione el origen de datos en la lista Data Sources.
- 2 Haga clic en la casilla **Log File Set Name**, modifique el nombre del conjunto de archivos de registro y haga clic en el botón **Set**.

Para agregar un nuevo origen de datos, siga estos pasos:

- 1 Haga clic en la casilla **Data Source Name** e introduzca un nuevo nombre.
- 2 Haga clic en la casilla **Log File Set Name**, introduzca un nuevo nombre completo del conjunto de archivos de registro y haga clic en el botón **Set**.

O bien

- 3 Haga clic en el botón **Browse** para seleccionar un origen de datos existente.

Para eliminar un origen de datos, siga estos pasos:

- 1 Seleccione el origen de datos en la lista Data Sources.
- 2 Haga clic en el botón **Delete**.
- 3 Cuando acabe de configurar el archivo de orígenes de datos, haga clic en **OK**.

Active los cambios:

Antes de continuar con otra tarea, *hay* que activar los cambios realizados en los orígenes de datos. Siga estos pasos:

- 1 Seleccione **Start/Stop** en el menú Agent de la ventana principal de Componente Performance Collection para abrir la ventana MeasureWare Services.
- 2 Haga clic en el botón **Stop Services** para detener los servicios de MeasureWare.
- 3 Cuando el botón **Stop Services** aparezca atenuado, haga clic en el botón **Start Services**.
- 4 Haga clic en el botón **Close** para volver a la ventana principal.

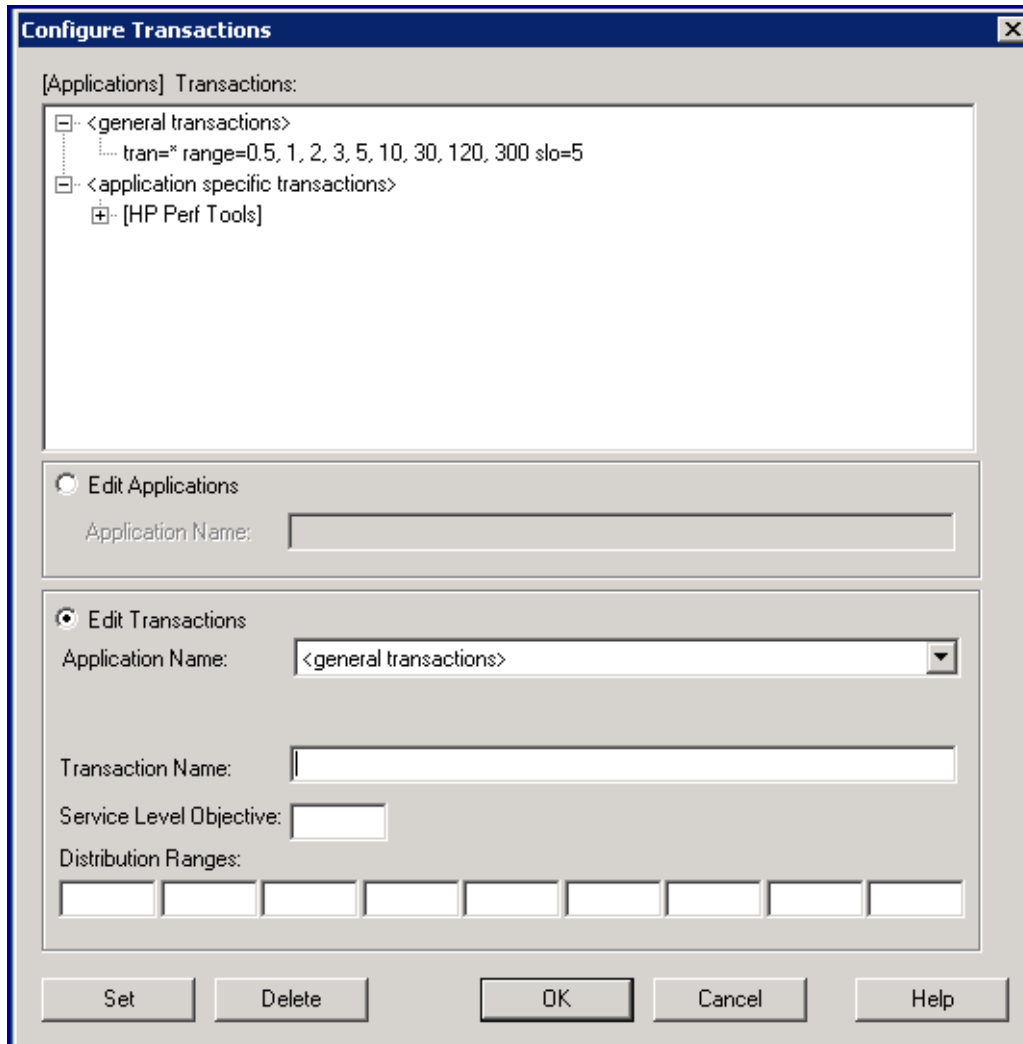
Para obtener instrucciones detalladas sobre cómo modificar el archivo de orígenes de datos, elija **Help Topics** en el menú de ayuda, seleccione "**How Do I...?**", y, a continuación, seleccione "**Modify a data source file**".

Configuración de transacciones

Se puede utilizar el archivo de configuración de transacciones, `ttdconf.mwc`, para personalizar la recopilación de datos de transacción para una aplicación. El archivo define el nombre de transacción, el rango de distribución del rendimiento y el objetivo de nivel de servicio que se desea cumplir en cada transacción. Opcionalmente, se pueden definir transacciones que son específicas para una aplicación.

El archivo `ttdconf.mwc` predeterminado consta de tres entradas. Dos entradas definen las transacciones usadas por el recopilador Componente Performance Collection `scopent` y una tercera, `tran=*`, registra todas las transacciones en las aplicaciones instrumentadas con las llamadas de función a la API de Application Response Measurement (ARM).

Figura 16 Cuadro de diálogo Configure Transactions



Si se están agregando nuevas aplicaciones al sistema que utilizan el objeto de nivel de servicio y los valores de rango desde la entrada `tran=*` en el archivo `ttdconf.mwc` predeterminado, no hay que hacer nada para incorporar las nuevas transacciones. Todos los valores predeterminados se aplican automáticamente a ellos.

Sin embargo, si se están agregando aplicaciones al sistema que tienen transacciones con sus propios objetivos de nivel de servicio y valores de rango de distribución, es preciso agregar estas transacciones al archivo `ttdconf.mwc`.



El orden de las entradas en el archivo `ttdconf.mwc` no es relevante. Se buscan primero las coincidencias exactas. Si no se encuentra ninguna, se utiliza la coincidencia mayor con un asterisco (*) al final.

Antes de realizar cualquier cambio en el archivo, consulte [Descripción del seguimiento de transacciones](#) on page 347 para obtener información sobre descripciones del formato de archivo de configuración, nombres de transacciones y aplicaciones, rangos de distribución de rendimiento y objetivos de nivel de servicio. Para realizar la configuración, haga clic en **Transactions**

en el menú **Configure** de la ventana principal de Componente Performance Collection para mostrar el cuadro de diálogo **Configure Transactions**. En este cuadro de diálogo, puede realizar las tareas siguientes:

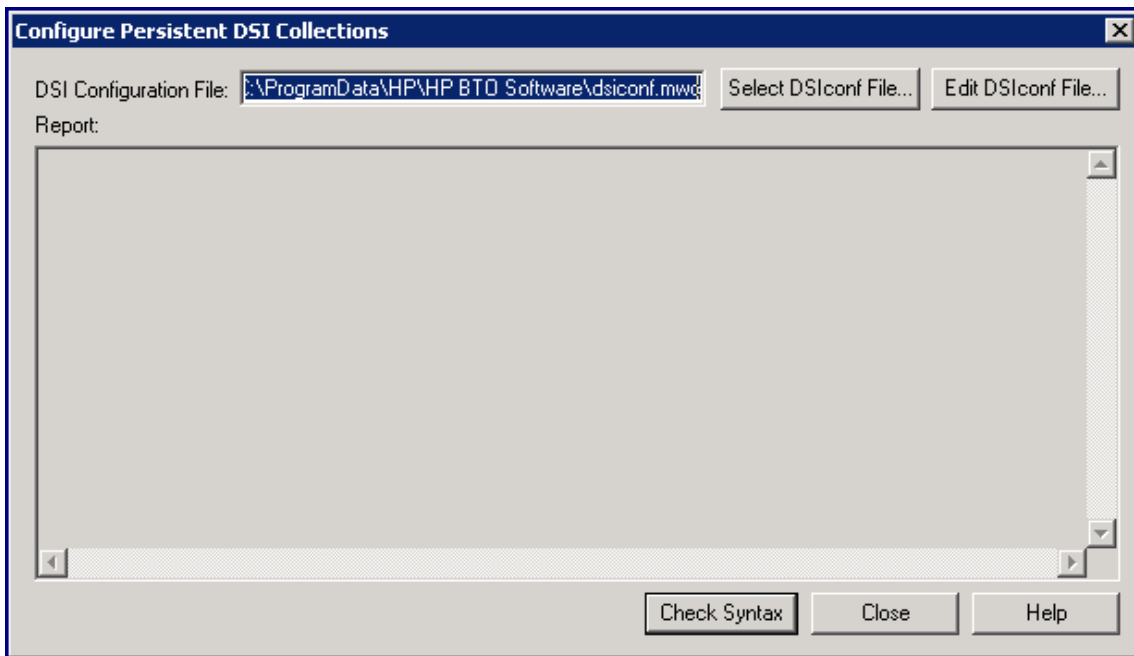
- 1 Agregar una transacción general
- 2 Agregar una transacción específica de la aplicación
- 3 Modificar un rango de distribución del rendimiento de la transacción u objetivo de nivel de servicio
- 4 Eliminar una transacción

Para obtener instrucciones detalladas sobre cómo realizar estas tareas, elija **Help Topics** en el menú de ayuda, seleccione "**How Do I...?**", y, a continuación, seleccione "**Configure transactions**".

Configuración de recopilaciones DSI persistentes

El comando **Persistent DSI Collections** del menú Configure se utiliza para comprobar la sintaxis o modificar el archivo de configuración DSI, `dsiconf.mwc`. El archivo `dsiconf.mwc` se utiliza para configurar el registro continuo de recopilaciones de datos que se incorporaron a Componente Performance Collection desde orígenes externos. Para obtener más información, consulte [Introducción general de Integración de Orígenes de Datos \(DSI\)](#) on page 263.

Figura 17 Cuadro de diálogo Configure Persistent DSI Collections



Para comprobar la sintaxis del archivo de configuración DSI, siga estos pasos:

- 1 Haga clic en **Persistent DSI Collections** en el menú Configure de la ventana principal de Componente Performance Collection. Aparecerá el cuadro de diálogo Configure Persistent DSI Collections mostrando el nombre del archivo `dsiconf.mwc` actualmente abierto.
- 2 Para comprobar otro archivo `dsiconf.mwc`, haga clic en el botón **Select DSIconf File**.
- 3 Para comprobar la sintaxis del archivo, haga clic en el botón **Check Syntax**. Los errores o advertencias resultantes se mostrarán en la ventana Report Viewer de Componente Performance Collection.
- 4 Para modificar cualquier parte del archivo, haga clic en el botón **Edit DSIconf File**. Puede colocar los cuadros de diálogo Edit DSIconf File y Configure Persistent DSI Collections en la pantalla para poder utilizar ambos al mismo tiempo.

Para obtener instrucciones detalladas sobre cómo comprobar la sintaxis del archivo de configuración DSI, elija **Help Topics** en el menú de ayuda, seleccione "How Do I...?", y a continuación, seleccione "**Check the syntax of a DSI configuration file**".

Para modificar un archivo de configuración DSI, siga estos pasos:

- 1 Haga clic en **Persistent DSI Collections** en el menú Configure de la ventana principal de Componente Performance Collection y, a continuación, haga clic en el botón **Edit DSIconf File** en el cuadro de diálogo Configure Persistent DSI Collections. Se mostrará el contenido

del archivo `dsiconf.mwc` actualmente abierto en el editor o procesador de texto previamente especificado. (Para especificar un editor o procesador de texto, consulte [Configuración de opciones de usuario](#) on page 243.)

- 2 Antes de realizar los cambios en el archivo, consulte [Uso de Performance Collection Component en Windows](#) on page 223 para obtener información sobre las convenciones que se deben seguir.
- 3 Modifique el archivo, si es preciso, y guárdelo en formato de texto.

Antes de continuar con otra tarea, *hay* que activar los cambios realizados en el archivo `dsiconf.mwc`. Siga estos pasos:

- 1 Haga clic en **Start/Stop** en el menú Agent de la ventana principal de Componente Performance Collection para abrir la ventana MeasureWare Services.
- 2 Seleccione la casilla **Persistent DSI Collections**.
- 3 Haga clic en el botón **Refresh**.
- 4 Haga clic en el botón **Close** para volver a la ventana principal.

Para obtener instrucciones detalladas sobre cómo modificar un archivo de configuración DSI, elija **Help Topics** en el menú de ayuda, seleccione "**How Do I...?**", y, a continuación, seleccione "**Modify a DSI configuration file**".



Si se utiliza WordPad, Notepad o Microsoft Word para modificar el archivo `dsiconf.mwc` y después se usa el comando **Save As** para guardarlo, se agregará automáticamente la extensión `.txt` predeterminada al nombre del archivo. Se dispondrá entonces de un archivo denominado `dsiconf.mwc.txt`. Para conservar el nombre de archivo `dsiconf.mwc`, se utiliza el comando **Save As** para guardar el archivo como un archivo de texto y se rodea el nombre de archivo entre comillas ("). Por ejemplo: "`dsiconf.mwc`"

Comprobación del estado de Componente Performance Collection

El comando **Status** del menú Agent se utiliza para revisar el estado actual de los procesos de Componente Performance Collection. La información la genera el programa `perfstat`.

Para indicar la información específica que se va a incluir en el informe de estado, se selecciona el comando **Options** en el menú Configure y se selecciona cualquiera de las opciones siguientes en el cuadro de diálogo Configure Options.

Running Processes

Se muestran los procesos en segundo plano y en primer plano que se están ejecutando actualmente en Componente Performance Collection. También se enumeran los procesos en segundo plano que deberían estar ejecutándose pero que *no* se ejecutan.

Datacomm Services

Los servicios de comunicación de datos se localizan y se comunican con los servicios de comunicación de datos de Componente Performance Collection. Muestran si el proceso de servidor de base de datos del generador de alarmas (agdbserver) está o no ejecutándose y respondiendo. Si no están habilitadas las comunicaciones de datos, esta información puede tardar más de 30 segundos en generarse mientras espera que los servicios de comunicación de datos respondan.

System Services

Se muestra el estado actual de los servicios del sistema de Componente Performance Collection, como el recopilador scope, el administrador de transacciones y la interfaz de medidas.

System Configuration

Nombre del sistema, versión del sistema operativo y tipo de procesador.

File Version Numbers

Números de versión de los archivos de Componente Performance Collection. Se indican todos los archivos críticos que faltan.

Status File Latest Entries

Las últimas entradas de cada archivo de estado de la herramienta de rendimiento.

Status File Warnings and Errors

Todas las líneas de los archivos de estado de la herramienta de rendimiento que contengan "Error" o "Warning". Se puede producir un listado muy grande en aquellos casos en que se hayan omitido las advertencias durante prolongados períodos de tiempo.

Para enumerar el estado actual, haga clic en **Status** en el menú Agent de la ventana principal de Componente Performance Collection. Report Viewer de Componente Performance Collection mostrará la información seleccionada en el cuadro de diálogo Configure Options.

Para obtener un informe completo de toda la información de estado, haga clic en **Report** en el menú Agent. Report Viewer de Componente Performance Collection mostrará una lista completa de toda la información de estado.

Para obtener instrucciones detalladas sobre cómo comprobar el estado de Componente Performance Collection, elija **Help Topics** en el menú de ayuda, seleccione "How Do I...?", y, a continuación, seleccione "**Check status of Performance Collection Component processes.**"

También se puede ejecutar el programa `perfstat` desde el Símbolo de sistema de Windows.

Creación de recopilación de contadores de rendimiento

Componente Performance Collection proporciona acceso a los contadores de rendimiento de Windows que se utilizan para medir el rendimiento del sistema, aplicaciones o dispositivos en el sistema. Se puede usar Extended Collection Builder and Manager (ECBM) para seleccionar contadores de rendimiento específicos para crear recopilaciones de datos.

Creación de una recopilación de contadores de rendimiento

Para crear una recopilación, seleccione **Extended Collections** en el menú Agent de la ventana principal de Componente Performance Collection. Aparecerá la ventana Extended Collection Builder and Manager, mostrando una lista de objetos de Windows en el panel izquierdo. Para obtener información sobre cómo crear recopilaciones, seleccione **Help Topics** en el menú de ayuda de la ventana Extended Collection Builder and Manager.

Después de crear las recopilaciones de contadores de rendimiento de Windows, se utiliza el panel Extended Collection Manager situado en la parte inferior para registrar, iniciar y detener recopilaciones tanto nuevas como existentes.

Administración de una recopilación de contadores de rendimiento

Para administrar las recopilaciones de datos, hay que utilizar el panel Extended Collection Manager situado en la parte inferior de Extended Collection Builder and Manager. Inicialmente no aparecerá ninguna recopilación porque hay que registrar una antes de comenzar a recopilar datos.

Una vez registrada o almacenada la recopilación que se ha creado, el panel Extended Collection Manager mostrará una lista de recopilaciones actuales. El panel Extended Collection Manager también muestra el estado de todas las recopilaciones y permite ver la información (propiedades) sobre la misma recopilación. Para obtener información sobre cómo administrar las recopilaciones, seleccione **Help Topics** en el menú **Help** de la ventana Extended Collection Builder and Manager.

Sugerencias para el uso de Extended Collection Builder and Manager

- El archivo `<Installdir>\paperdocs\mwa\C\monxref.txt` contiene una referencia cruzada de métricas de Componente Performance Collection para los comandos y contadores de rendimiento de Windows. El registro de datos a través de Extended Collection Builder and Manager para métricas ya recopiladas por Componente Performance Collection ocasiona una sobrecarga adicional en el sistema.
- Cuando se utiliza Extended Collection Builder para crear recopilaciones, se asignan nombres de métricas predeterminados a los contadores de rendimiento de Windows para su uso interno con Componente Performance Collection. Estos nombres predeterminados no suelen ser descriptivos ni fáciles de descifrar. Para hacer más descriptivos estos nombres de métricas o para hacerlos coincidir con los nombres de métricas proporcionados por la aplicación origen, hay que modificar los atributos de las métricas haciendo clic con el botón derecho o doble clic en el nombre de la métrica después de haberlo arrastrado del panel izquierdo al derecho en la ventana Extended Collection Builder and Manager. (Consulte la ayuda en línea de Extended Collection Builder and Manager para obtener información detallada.)

- Si se comienzan con 61 o más recopilaciones, las posteriores a la recopilación 60 producirán estados de error. Esto puede causar problemas con otras recopilaciones.
- Si se recopilan métricas de disco lógico desde un sistema configurado con **Wolfpack**, hay que reiniciar la recopilación para recopilar datos para cualquier nueva instancia del disco que no esté presente cuando se registró la recopilación.
- La eliminación correcta de las recopilaciones requiere el reinicio de Componente Performance Collection después de eliminar la recopilación. Si no se reinicia Componente Performance Collection, es posible que se produzca un error durante la operación de eliminación. Este error suele significar que algunos archivos no se eliminaron correctamente. Es posible que haya que eliminar manualmente los archivos y directorios que queden después de reiniciar Componente Performance Collection.
- Extended Collection Builder and Manager puede informar de valores que faltan para algunas métricas con contadores en caché. Este problema puede ocurrir en determinadas circunstancias cuando se produce un desbordamiento en un valor de métrica. También se envía un mensaje al archivo de estado ECBM. Este problema se soluciona reiniciando la recopilación.

En la ayuda en línea de Extended Collection Builder and Manager se podrá obtener información sobre conceptos e instrucciones sobre la creación y visualización de recopilaciones de datos en Extended Collection Builder and Manager. Para ver la ayuda en línea, desde el escritorio, seleccione **Inicio** → **Programas** → **HP** → **Operations Agent** → **Performance Collection Component** → **ECB-ECM Online Help**. Se puede seleccionar **Extended Collections** del menú **Agent** en la ventana principal de Componente Performance Collection y seleccionar **Help Topics** en el menú de ayuda de la ventana Extended Collection Builder and Manager. La ayuda en línea está disponible seleccionando el botón **Help** en los cuadros de diálogo que aparecen en Extended Collection Builder and Manager.

Administración de ECBM desde la línea de comandos

Se puede ejecutar el programa ECBM desde el directorio `<rpmtools>\bin` mediante el Símbolo de sistema de Windows.

Las recopilaciones se pueden administrar desde la línea de comandos utilizando el comando siguiente:

```
\rpmtools\bin\mwcmcmd.exe
```

Para mostrar varias opciones, escriba el comando siguiente:

```
\rpmtools\bin\mwcmcmd /?
```

Para iniciar las recopilaciones detenidas, escriba el comando siguiente:

```
mwcmcmd start <collection_name(s)>
```

Para iniciar una nueva recopilación desde una directiva de instancias de variables, escriba el comando siguiente:

```
mwcmcmd start <policy_name> <collect_name> <instance(s)> [options]
```

Están disponibles las siguientes opciones:

```
-i <sampling_interval> - change sampling interval (seconds)
-l <logfile_path_name> - change default log location
-a <alarm_file>         - change the alarm definitions file
```

Para detener las recopilaciones activas, escriba el comando siguiente:

```
mwcmcmd stop <collection_name(s)>
```

A continuación se muestra el comando para registrar un archivo de directiva:

```
mwcmcmd register <policy_file> <collection/policy_name> [options]
```

Las opciones siguientes sólo están disponibles al registrar un archivo de directiva de instancia fija:

```
-i <sampling_interval> - change sampling interval (seconds)
-l <logfile_path_name> - change default log location
-a <alarm_file>         - change the alarm definitions file
```

Para eliminar una única recopilación:

```
mwcmcmd delete <collection/policy_name> [options]
```

Las opciones siguientes sólo están disponibles al eliminar una recopilación:

```
-p <archive_path> - archives logfiles to specified path
-r                - restarts Performance agent
```

Para eliminar varias recopilaciones o directivas:

```
mwcmcmd delete { <collection/policy_name(s)> | -c | -all }
-c          - deletes ALL collections
-a          - deletes ALL collections and policies
```



Al eliminar más de una directiva/recopilación a la vez, Componente Performance Collection se reiniciará automáticamente y se eliminarán todos los archivos de registro asociados.

Para enumerar todas las recopilaciones y directivas registradas, escriba el comando siguiente:

```
mwcmcmd list
```

Para enumerar todas las propiedades de una recopilación o directiva, escriba el comando siguiente:

```
mwcmcmd properties <collection/policy_name>
```

Para enumerar los objetos de instancias de variables en una directiva, escriba:

```
mwcmcmd objects <policy_name>
```

12 Introducción general de Integración de Orígenes de Datos (DSI)

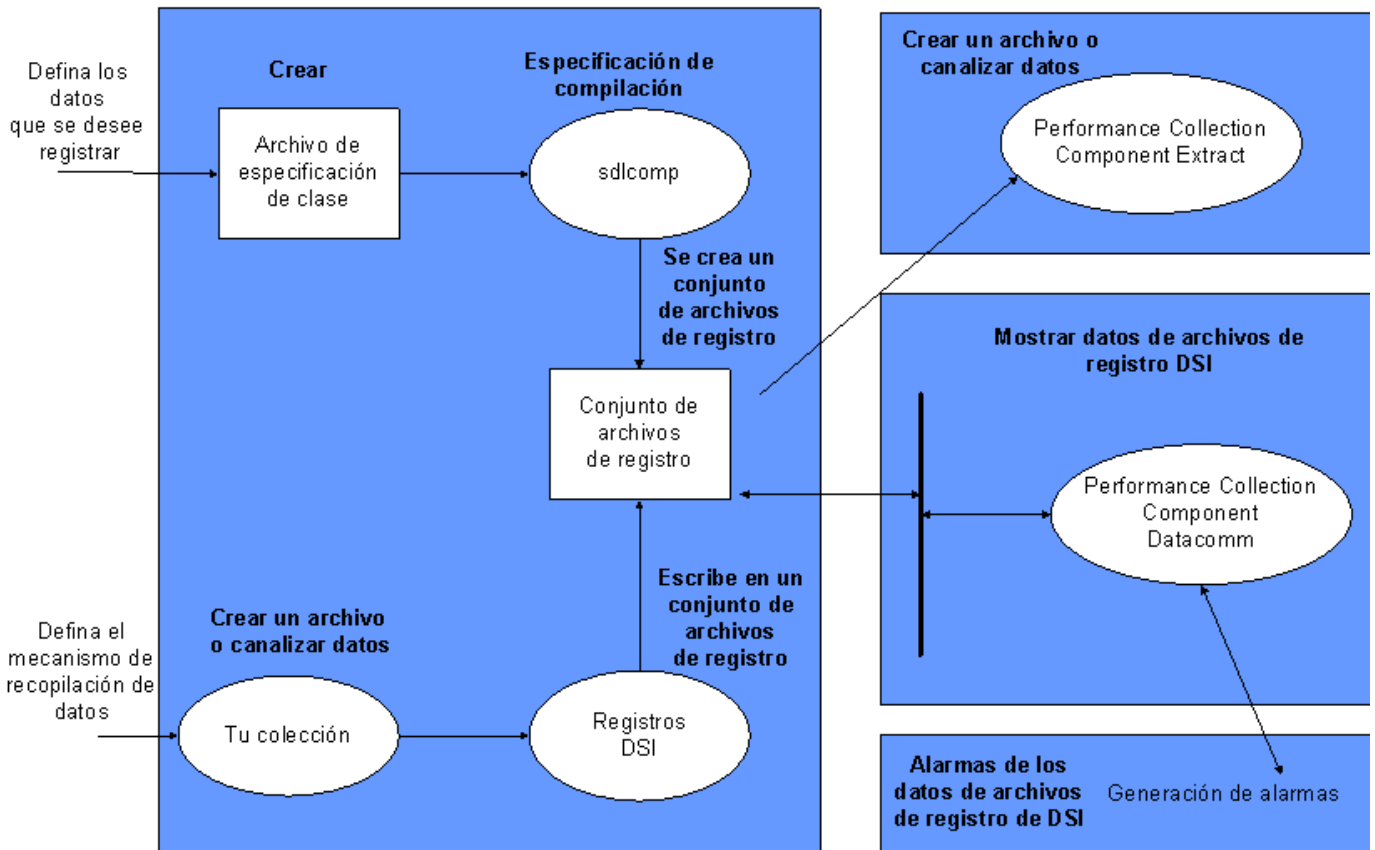
La tecnología de integración de orígenes de datos (DSI) permite usar HP Operations Agent para registrar datos, definir alarmas y acceder a métrica de orígenes de datos distintos de las métricas registradas por el colector `scope` de Componente Performance Collection. La métrica puede proceder de orígenes de datos como bases de datos, monitores LAN y aplicaciones de usuarios finales.

Los datos que se registran con DSI pueden mostrarse en HP Performance Manager junto con la métrica de rendimiento estándar registrada por el recopilador `scope`. Los datos registrados DSI pueden ser exportados con el programa `extract` de Componente Performance Collection para ser mostrados en hojas de cálculo o paquetes de análisis similares.

Funcionamiento de DSI

El diagrama siguiente muestra cómo crear y usar archivos de registro DSI para registrar y gestionar datos. Los archivos de registro DSI contienen datos autodescription que son recopilados fuera del recopilador `scope` de Componente Performance Collection. Los procesos DSI se describen en mayor detalle en la página siguiente.

Figura 18 Proceso de integración de orígenes de datos
Proceso de integración de orígenes de datos



Usar DSI para registrar datos comporta las siguientes tareas:

Creación de la especificación de clase

En primer lugar es preciso crear y compilar una especificación por cada clase de datos que se desee registrar. La especificación describe la clase de datos y la métrica individual que será registrada en la clase. Al compilar la especificación con el compilador DSI `sdlcomp`, se creará un conjunto de archivos de registro vacíos para aceptar datos del programa `dsilog`. Este proceso crea un conjunto de archivos de registro que contiene un archivo raíz, un archivo de descripción y uno o más archivos de datos.

Recopilación y archivo de datos

A continuación el usuario recopilará los datos que desea registrar iniciando el proceso correspondiente. El usuario podrá o bien canalizar el resultado del proceso de recopilación al programa `dsilog` directamente o de un archivo en el que estuvieran almacenados los datos. `dsilog` procesa los datos según la especificación y los escribe en el archivo de registro pertinente. `dsilog` permite al usuario especificar la forma y el formato de los datos de entrada.

Los datos introducidos en el proceso DSI deberían contener registros de datos múltiples. Un registro consta de valores de métrica contenidos en una única línea. Si se envían datos de registro en registro a DSI, detenga el proceso y a continuación envíe otro registro, `dsilog` puede anexar pero no resumir los datos.

Uso de datos

Performance Manager permite al usuario mostrar datos de archivos de registro DSI. También puede usarse el programa `extract` de Componente Performance Collection para exportar los datos para su uso con otras herramientas de análisis. Así mismo, es posible configurar alarmas que ocurran cuando la métrica DSI exceda las condiciones definidas.

13 Uso de integración de orígenes de datos

El presente capítulo incluye una introducción general de cómo usar la tecnología DSI y contiene la información siguiente:

- Planificación de recopilación de datos
- Definición del formato de archivo de registro en el archivo de especificación de clase
- Creación de un conjunto de archivos de registro vacíos
- Archivo de datos en el conjunto de archivos de registro
- Uso de los datos registrados

Para información de referencia detallada sobre especificaciones de clases DSI y programas DSI, consulte [Capítulo 14, Referencia de especificación de clase DSI](#) y [Capítulo 15, Referencia del programa DSI](#).

Planificación de recopilación de datos

Antes de crear los archivos de especificación de clase DSI e iniciar el proceso de registro, es necesario adquirir conocimiento sobre los temas siguientes:

- Estar bien familiarizado con el entorno de manera que se sepa qué clases de datos son de utilidad para la administración de recursos informáticos.
- ¿Qué datos se encuentran disponibles?
- ¿Dónde están los datos?
- ¿Cómo pueden recopilarse los datos?
- ¿Cuáles son los delimitadores entre ítems de datos? Para un correcto procesamiento de `dsilog`, los valores de métrica del flujo de entrada deberán estar separados por espacios en blanco (predeterminado) o un delimitador definido por el usuario.
- ¿Cuál es la frecuencia de recopilación?
- ¿Cuánto espacio se requiere para mantener registros?
- ¿Cuál es la salida del programa o proceso que usa el usuario para acceder a los datos?
- ¿Qué alarmas se desean generar y bajo qué condiciones?
- ¿De qué opciones se dispone para el registro con la especificación de clase y el proceso `dsilog`?

Definición del formato de archivo de registro

Una vez tenga una clara visión sobre el tipo de datos que desea recopilar, cree una especificación de clase para definir los datos a ser registrados y el conjunto de archivos de registro que contendrá los datos registrados. Introduzca la información siguiente en la especificación de clase:

- Nombre de clase de datos y número de ID
- Nombre de etiqueta (opcional) que es un sustituto del nombre de clase. (Por ejemplo, si existe un nombre de etiqueta, podrá ser usado en Performance Manager).
- Lo que desea que ocurra cuando los datos antiguos sean suprimidos para hacer sitio a los nuevos. Consulte [Modo en que están organizados los archivos de registro](#) para obtener más información.
- Los nombres de métrica y demás información descriptiva, como la cantidad de decimales permitidos para los valores de métrica.
- El modo en que desea que se resuman los datos si se desea registrar un número limitado de registros por hora.

A continuación se muestra un ejemplo de una especificación de clase:

```
CLASS VMSTAT_STATS = 10001
LABEL "VMSTAT data"
INDEX BY HOUR
MAX INDEXES 12
ROLL BY HOUR
RECORDS PER HOUR 120;

METRICS
RUN_Q_PROCS      = 106
LABEL "Procs in run q"
PRECISION 0;

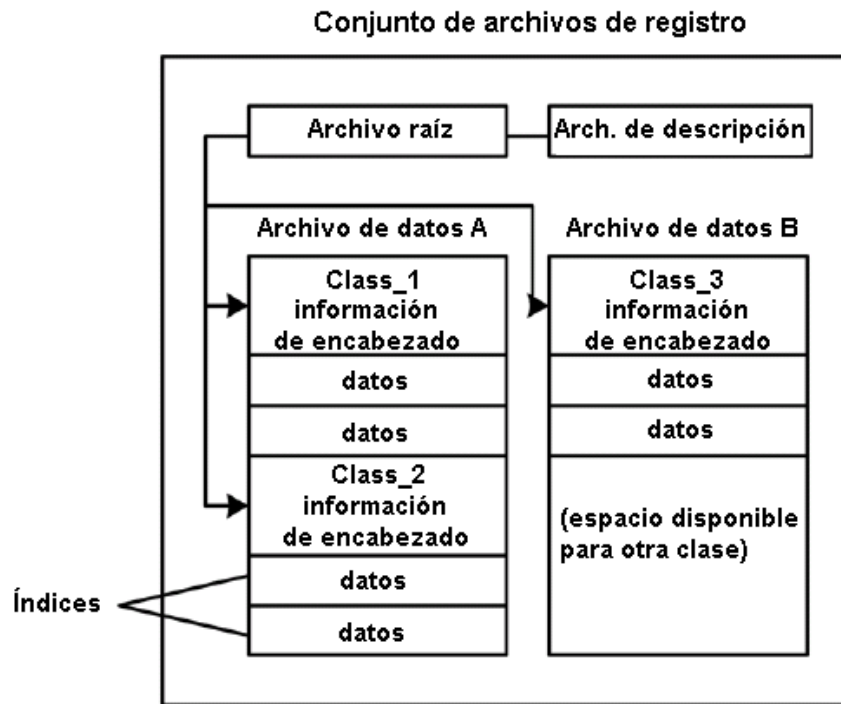
BLOCKED_PROCS    = 107
LABEL "Blocked Processes"
PRECISION 0;
```

En un archivo de especificación de clase se puede incluir una o múltiples clases. Cuando finalice el archivo de especificación de clase, asigne un nombre al archivo y guárdelo. Al ejecutar el compilador DSI `sdlcomp`, use este archivo para crear el conjunto de archivos de registro. Para más información sobre sintaxis de descripción de métrica y especificaciones de clase, consulte [Capítulo 14, Referencia de especificación de clase DSI](#).

Modo en que están organizados los archivos de registro

Los archivos de registro están organizados en clases. Cada clase representa un origen de datos entrantes y consta de un grupo de elementos de datos o métrica que se registran conjuntamente. Cada registro o fila de datos de una clase representa un ejemplo de los valores correspondientes a ese grupo de métrica.

Los datos de clases son almacenados en disco en archivos de registro que forman parte del conjunto de archivos de registro. El conjunto de archivos de registro contiene un archivo raíz, un archivo de descripción y uno o más archivos de registro. Todos los datos de una clase son siempre guardados en un mismo archivo de datos. Sin embargo, cuando se proporciona un nombre de conjunto de archivos de registro al compilador `sd1comp`, se pueden almacenar múltiples clases juntas en un mismo conjunto de archivos de registro o en conjuntos distintos de archivos de registro. La figura siguiente muestra cómo almacenar dos clases en un mismo conjunto de archivos de registro.



Dado que cada clase se crea como archivo de registro circular, se puede establecer la capacidad de almacenamiento de cada clase por separado, incluso si se ha especificado que se deben almacenar múltiples clases en un mismo conjunto de archivos de registro. Cuando se alcance la capacidad de almacenamiento, la clase será “rolled” (suprimida), lo que significa que los registros más antiguos de la clase serán eliminados para hacer sitio a nuevos datos.

Se puede especificar que se realice una acción cuando la clase sea suprimida, como por ejemplo exportar los datos antiguos a un archivo de registro.

Creación del conjunto de archivos de registro

El compilador DSI `sdlcomp`, usa el archivo de especificación de clase para crear o actualizar un conjunto de archivos de registro vacíos. El conjunto de archivos de registro es entonces usado para recibir datos registrados del programa `dsilog`.

Para crear un conjunto de archivos de registro siga los pasos siguientes:

- 1 Ejecute `sdlcomp` con las variables y opciones adecuadas. Por ejemplo:

```
sdlcomp [-maxclass value] specification_file  
[logfile_set[log file]] [options]
```

- 2 Compruebe el resultado por si se han producido errores e introduzca los cambios necesarios.

Para obtener más información sobre `sdlcomp`, consulte la [Sintaxis del compilador](#) en Capítulo 15.

Pruebas del archivo especificación de clase y el proceso de registro (Opcional)

DSI usa un programa, `sdlgendata`, que permite al usuario probar el archivo de especificación de clase con un origen de entrada de datos generados. El usuario podrá examinar el resultado del proceso para verificar si DSI puede registrar datos según sus especificaciones. Para obtener más información sobre `sdlgendata`, consulte [Pruebas del proceso de registro con Sdlgendata](#) en Capítulo 15.)

Para probar su archivo de especificación de clase para el proceso de registro:

- 1 Suministre los datos generados por `sdlgendata` al programa `dsilog`. La sintaxis es la siguiente:

```
sdlgendata logfile_set class | dsilog logfile_set class -vo
```

- 2 Compruebe el resultado para ver si el archivo de especificación de clase coincide con el formato de su proceso de recopilación de datos. Si el programa `sdlgendata` da como resultado algo distinto a su programa, se ha producido un error o bien en el formato de salida o en el archivo de especificación de clase.
- 3 Antes de comenzar a recopilar datos reales, elimine todos los archivos de registro del proceso de pruebas.

Archivo de datos en el conjunto de archivos de registro

Una vez que haya creado el conjunto de archivos de registro y los haya probado (opcional), actualice los archivos de configuración de Componente Performance Collection según se requiera y a continuación ejecute el programa `dsilog` para registrar datos de entrada.

- 1 Actualice el archivo de configuración de origen de datos, `datasources`, para agregar los archivos de registro DSI como orígenes de datos para generar alarmas.
- 2 Modifique el archivo de definiciones de alarma, `alarmdef`, si desea generar alarmas con métricas DSI específicas. Para obtener más información, consulte [Definición de alarmas en métrica DSI](#) en Capítulo 15.
- 3 De manera opcional, pruebe el proceso de registro canalizando los datos (que pueden ser generados por `sdlgendata` para hacer coincidir su especificación de clase) al programa `dsilog` con el conjunto de opciones `-vi`.
- 4 Compruebe los datos para asegurarse de que estén siendo registrados correctamente.
- 5 Después de realizar las pruebas, elimine los datos que hayan sido probados.
- 6 Inicie el proceso de recopilación en la línea de comandos.
- 7 Canalice los datos del proceso de recopilación al archivo `dsilog` (o usando otro método que los lleve a `stdin`) con los conjuntos de opciones y las variables adecuadas. Por ejemplo:

```
<program or process with variables>| dsilog logfile_set class
```



El programa `dsilog` ha sido diseñado para recibir un flujo de datos continuo. Por lo tanto, es importante estructurar las secuencias de comandos de manera que `dsilog` reciba datos de entrada continuos. No escriba secuencias que creen un nuevo proceso `dsilog` para los nuevos puntos de datos de entrada. Ello podría provocar que se escriban marcas de tiempo dobles en el archivo `dsilog`, y puede ocasionar problemas en Performance Manager y `perfalarm` a la hora de leer el archivo. Consulte el [Capítulo 16, Ejemplos de integración de orígenes de datos \(DSI\)](#) para obtener ejemplos sobre secuencias recomendadas y problemáticas.

Para obtener más información sobre las opciones `dsilog`, consulte el [Proceso de registro dsilog](#) en Capítulo 15.

Uso de los datos registrados

Una vez que haya creado los archivos de registro DSI podrá exportar los datos con el programa `extract` de Componente Performance Collection'. también es posible configurar alarmas que ocurran cuando la métrica DSI exceda las condiciones definidas.

A continuación se indican varios modos de usar datos DSI registrados:

- Exportar los datos para su uso con herramientas de informes como hojas de cálculo.
- Mostrar datos DSI exportados con herramientas de análisis como Performance Manager.
- Monitorizar alarmas con HP Operations Manager o HP Network Node Manager.



No se pueden crear archivos de registro extraídos de los archivos de registro DSI.

14 Referencia de especificación de clase DSI

El presente capítulo ofrece información de referencia detallada sobre:

- Especificaciones de clase
- Sintaxis de especificaciones de clase
- Descripciones de métrica en las especificaciones de clase

Especificaciones de clase

Por cada origen de datos de entrada será preciso crear un archivo de especificación de clase para describir el formato de almacenamiento de los datos de entrada. Para crear el archivo, use el lenguaje de especificación de clase descrito en la sección siguiente, [Sintaxis de especificación de clase](#). El archivo de especificación de clase contiene:

- una descripción de clase, la cual asigna un nombre y un ID numérico al conjunto de datos de entrada, determina la cantidad de datos que serán almacenados y especifica cuándo se suprimirán datos para hacer sitio a otros nuevos.
- descripciones de métrica de cada ítem de datos. Las descripciones de métrica asignan un nombre al ítem de datos y lo describen. También especifica el nivel de resumen que se debe aplicar a los datos (RECORDS PER HOUR) si más de un registro se recibe en el intervalo de tiempo que ha sido configurado para la clase.

Para generar el archivo de especificación de clase, use cualquier editor o procesador de textos que le permita guardar el archivo como archivo de texto ASCII. Para especificar el nombre del archivo de especificación de clase será preciso ejecutar `sdlcomp` para compilarlo. Cuando la especificación de la clase haya sido compilada, se creará o actualizará de manera automática un conjunto de archivos de registro para almacenar los datos.

La especificación de clase permite determinar cuántos registros por hora serán almacenados para una clase y especificar el método de resumen que deberá ser usado si se reciben más registros de los que se desean almacenar. Por ejemplo, si ha solicitado que se almacenen 12 registros por hora (un registro cada cinco minutos) y se reciben registros cada minuto, podría tener promediados algunos de los ítems de datos y otros totalizados para mantener un recuento en ejecución.



El compilador DSI, `sdlcomp`, crea archivos con los nombres siguientes para un conjunto de archivos de registro (llamado `logfile_set_name`):

```
logfile_set_name y logfile_set_name.desc
```

`sdlcomp` crea un archivo con el nombre predeterminado siguiente para una clase (llamado `class_name`):

```
logfile_set_name.class_name
```

Evite el uso de nombres de archivos de especificación de clase que entren en conflicto con estas convenciones de nomenclatura, o de lo contrario `sdlcomp` no funcionará correctamente.

Sintaxis de especificación de clase

Las instrucciones de sintaxis que se muestran entre corchetes [] son opcionales. Instrucciones múltiples entre llaves { } indican que una de las instrucciones debe ser seleccionada. Las palabras en cursiva indican un número o nombre de variable introducido por el usuario. Las comas pueden utilizarse en cualquier lugar para separar instrucciones de sintaxis si ello supone mayor claridad, excepto directamente antes de un punto y coma, lo cual marca el final de la especificación de clase y el final de cada especificación de métrica. Las instrucciones no distinguen entre mayúscula y minúscula.



Las descripciones definidas por el usuario, como por ejemplo *metric_label_name* o *class_label_name*, no pueden coincidir con ningún elemento de palabra clave de la sintaxis de especificación de clase DSI.

Los comentarios empiezan por # o //. Todo lo que siga a # o // en una misma línea será ignorado. Adviértase el punto y coma requerido después de la descripción de clase y de cada descripción de métrica. A continuación se ofrece información detallada sobre cada parte de la especificación de clase y diversos ejemplos:

```
CLASS class_name = class_id_number
[LABEL "class_label_name"]

[INDEX BY {HOUR | DAY | MONTH} MAX INDEXES number
[[ROLL BY {HOUR | DAY | MONTH} [ACTION "action" ]
[ CAPACITY {maximum_record_number} ]
[ RECORDS PER HOUR number ]
;

METRICS

metric_name = metric_id_number
[ LABEL "metric_label_name" ]
[ TOTALED | AVERAGED | SUMMARIZED BY metric_name ]
[ MAXIMUM metric_maximum_number ]
[PRECISION {0 | 1 | 2 | 3 | 4 | 5} ]
[TYPE TEXT LENGTH "length"]
;
```

Descripción de CLASE

Para crear una descripción de clase, asigne un nombre a un grupo de métrica de un origen de datos determinado, especifique la capacidad de la clase y designe el modo en que se suprimirán los datos de la clase cuando se exceda la capacidad.

Deberá comenzar la descripción de clase con la palabra clave `CLASS`. El parámetro final de la especificación de clase deberá estar seguido por un punto y coma.

Sintaxis

```
CLASS class_name = class_id_number

[LABEL "class_label_name"]

[INDEX BY { HOUR | DAY | MONTH } MAX INDEXES number
[ROLL BY { HOUR | DAY | MONTH } [ACTION "action"]

[ CAPACITY {maximum_record_number} ]
[ RECORDS PER HOUR number ]
;
```

Configuración predeterminada

La configuración predeterminada de la descripción de clase será:

```
LABEL (class_name)
INDEX BY DAY
MAX INDEXES 9
RECORDS PER HOUR 12
```

Para usar los valores predeterminados, introduzca sólo la palabra clave `CLASS` con un *class_name* y un *class_id_number* numérico.

CLASE

El nombre de clase e ID de clase identifican un grupo de métrica de un origen de datos específico.

Sintaxis

```
CLASS class_name = class_id_number
```

Instrucciones sobre su uso

class_name y *class_ID_number* deberán cumplir los requisitos siguientes:

- *class_name* es un dato alfanumérico y puede contener hasta 20 caracteres. El nombre puede empezar por un carácter alfabético y contener guiones bajos (si bien no caracteres especiales).
- *class_ID_number* deberá ser un dato numérico y contener un máximo de 6 dígitos.

- Ni *class_name* ni *class_ID_number* distinguen entre mayúscula y minúscula.
- *class_name* y *class_ID_number* deberán ser únicos entre las clases definidas por el usuario y no podrán coincidir con ninguna aplicación definida en el archivo `parm` de Componente Performance Collection. (Para obtener información sobre el archivo `parm` consulte el capítulo 2 de *HP Operations Agent correspondiente al Manual del usuario UNIX*).

Ejemplo

```
CLASS VMSTAT_STATS = 10001;
```

ETIQUETA

La etiqueta de clase identifica la clase en su totalidad. Es usada en lugar del nombre de clase en Performance Manager.

Sintaxis

```
[ LABEL "class_label_name" ]
```

Instrucciones sobre su uso

class_label_name deberá cumplir los requisitos siguientes:

- Deberá estar entre comillas dobles.
- Deberá contener un máximo de hasta 48 caracteres.
- No podrá coincidir con ninguno de los elementos de palabra clave de la sintaxis de especificación de clase DSI, como por ejemplo `CAPACITY` o `ACTION`, entre otros.
- Si contiene dobles comillas, anteponga una barra diagonal inversa (`\`). Por ejemplo, debería introducir `"\my\ data"` si la etiqueta fuese `"my data"`.
- Si no se ha especificado ninguna etiqueta se usará como predeterminado *class_name*

Ejemplo

```
CLASS VMSTAT_STATS = 10001
LABEL "VMSTAT data";
```

Parámetros INDEX BY, MAX INDEXES y ROLL BY

Los parámetros `INDEX BY`, `MAX INDEXES`, y `ROLL BY` permiten al usuario determinar cómo almacenar datos y cuándo suprimirlos. Con estos parámetros se designan los bloques de datos que deberán ser almacenados, el número máximo de bloques a almacenar y el tamaño de los bloques de datos que serán suprimidos cuando los datos alcancen su valor de índice máximo.

Sintaxis

```
[INDEX BY {HOURL | DAY | MONTH} MAX INDEXES number]
[[ROLL BY {HOURL | DAY | MONTH} [ACTION "action"]]
```

Instrucciones sobre su uso

El parámetro `INDEX BY` permite suprimir bloques de datos de la clase cuando se alcanza la capacidad de la clase. Las opciones `INDEX BY` y `RECORDS PER HOUR` podrán ser utilizadas para establecer indirectamente la capacidad de la clase como se describe a continuación en [Control del tamaño del archivo de registro](#).

El parámetro INDEX BY no puede superar el parámetro ROLL BY. Por ejemplo, INDEX BY DAY no funciona con ROLL BY HOUR, pero INDEX BY HOUR sí funciona con ROLL BY DAY.

Si el parámetro ROLL BY no ha sido especificado, se usará el parámetro INDEX BY. Cuando se alcance la capacidad se liberarán todos los registros contenidos en el intervalo de supresión más antiguo para su reutilización.

Toda ACTION especificada será realizada antes de que se supriman los datos. Esta ACTION opcional puede usarse para exportar los datos a otra ubicación antes de que sean suprimidos de la clase. Para obtener información sobre cómo exportar datos consulte el [Capítulo 15, Referencia del programa DSI](#).

Advertencia sobre las acciones de supresión

El comando UNIX especificado en la instrucción ACTION no puede ser ejecutado de fondo. No especifique tampoco un comando en la instrucción ACTION que cause un notable retraso puesto que los nuevos datos no serán registrados durante dicho retraso.

Si el comando es mayor de una línea, marque el inicio y fin de cada línea con dobles comillas. Asegúrese de incluir espacios donde sea necesario dentro de las comillas para que las diversas opciones de línea de comandos permanezcan separadas cuando se concadenen las líneas.

Si el comando contiene dobles comillas, anteponga una barra diagonal inversa (\).

La instrucción ACTION no podrá contener más de 199 caracteres.

En la instrucción ACTION pueden usarse macros para definir la ventana de tiempo de los datos que serán suprimidos del archivo de registro. Estos macros son expandidos por dsilog. Puede usar \$PT_START\$ para especificar el principio del bloque de datos que será suprimido en hora UNIX (segundos a partir de 1/1/70 00:00:00) y \$PT_END\$ para especificar el final de los datos en hora UNIX. Estos comandos resultan particularmente útiles cuando se combinan con el programa extract para exportar los datos antes de que sean sobrescritos.

Con macros se usará la longitud expandida con la limitación de caracteres de 199.

Ejemplos

Los siguientes ejemplos tienen por objeto clarificar la relación entre las cláusulas INDEX BY, MAX INDEXES y ROLL BY.

El siguiente ejemplo establece indirectamente la CAPACITY en 144 registros (1*12*12).

```
CLASS VMSTAT_STATS = 10001
LABEL "VMSTAT data"
INDEX BY HOUR
MAX INDEXES 12
RECORDS PER HOUR 12;
```

El siguiente ejemplo establece indirectamente la CAPACITY en 1440 registros (1*12*120).

```
CLASS VMSTAT_STATS = 10001
LABEL "VMSTAT data"
INDEX BY HOUR
MAX INDEXES 12
RECORDS PER HOUR 120;
```

El siguiente ejemplo muestra ROLL BY HOUR.

```
CLASS VMSTAT_STATS = 10001
LABEL "VMSTAT data"
INDEX BY HOUR
MAX INDEXES 12
ROLL BY HOUR
RECORDS PER HOUR 120;
```

El siguiente ejemplo muestra cómo todos los datos actualmente identificados para ser suprimidos (excluidos los fines de semana) son exportados a un archivo llamado `sys.sdl` antes de que los datos son sobrescritos. Adviértase que las últimas líneas del último ejemplo se encuentran entre comillas dobles para indicar que conforman un comando único.

```
CLASS VMSTAT_STATS = 10001
LABEL "VMSTAT data"
INDEX BY HOUR
MAX INDEXES 12
ROLL BY HOUR
ACTION "extract -xp -l sdl_new -C SYS_STATS "
"-B $PT_START$ -E $PT_END$ -f sys.sdl, purge -we 17 "
RECORDS PER HOUR 120;
```

Otros ejemplos

Los parámetros de índice sugeridos a continuación son útiles para determinar cuántos datos se desean almacenar.

INDEX BY	MAX INDEXES	Cantidad almacenada de datos
HOUR	72	3 días
HOUR	168	7 días
HOUR	744	31 días
DAY	365	1 año
MONTH	12	1 año

la siguiente tabla ofrece una explicación detallada de la configuración de ROLL BY

INDEX BY	MAX INDEXES	ROLL BY	Significado
DAY	9	DAY	9 días de datos serán almacenados en el archivo de registro. Antes de registrar el día 10, el día 1 será suprimido. Estos son los valores predeterminados de index y max indexes.
HOUR	72	HOUR	72 horas de datos (3 días) serán almacenadas en el archivo de registro. Antes de registrar la hora 73, la hora 1 será suprimida. En adelante, al comenzar cada hora subsiguiente, la hora "más antigua" será suprimida.
HOUR	168	DAY	168 horas (7 días) de datos serán almacenadas en el archivo de registro. Antes de registrar la hora 169 (día 8), el día 1 será suprimido. En adelante, al comenzar cada día subsiguiente, el día "más antiguo" será suprimido.
HOUR	744	MONTH	744 horas de datos (31 días) serán almacenados en el archivo de registro. Antes de registrar la hora 745 (día 32), el mes 1 será suprimido. En adelante, antes de registrar la hora 745, el mes "más antiguo" será suprimido. Por ejemplo, el archivo <code>dsilog</code> es iniciado el día 15 de abril y registra datos hasta el 16 de mayo (744 horas). Antes de registrar la hora 745 (la primera hora del 17 de mayo), <code>dsilog</code> suprimirá los datos del mes de abril (15-30 de abril).

INDEX BY	MAX INDEXES	ROLL BY	Significado
DAY	30	DAY	<p>30 días de datos serán almacenados en el archivo de registro. Antes de registrar el día 31, el día 1 será suprimido. En adelante, al comenzar cada día subsiguiente, el mes "más antiguo" será suprimido.</p> <p>Por ejemplo, el archivo <code>dsilog</code> es iniciado el 1 de abril y registra datos todo el mes, así que el 1 de abril será suprimido cuando los datos del 1 de mayo (día 31) sean registrados.</p>
DAY	62	MONTH	<p>62 días de datos serán almacenados en el archivo de registro. Antes de registrar el día 63, el mes 1 será suprimido. En adelante, antes de registrar el día 63, el mes "más antiguo" será suprimido.</p> <p>Por ejemplo, si el archivo <code>dsilog</code> es iniciado el 1 de marzo y archiva datos para los meses de marzo y abril, habrán 61 días de datos en el archivo de registro. Una vez que <code>dsilog</code> archive los datos del 1 de mayo (el día 62), el archivo de registro estará lleno. Antes de que el archivo <code>dsilog</code> registre los datos del 2 de mayo, suprimirá todo el mes de marzo.</p>

INDEX BY	MAX INDEXES	ROLL BY	Significado
MONTH	2	MONTH	<p>2 meses de datos serán almacenados en el archivo de registro. Antes de registrar el tercer mes, el mes 1 será suprimido. En adelante, al comenzar cada mes subsiguiente, el mes "más antiguo" será suprimido.</p> <p>Por ejemplo, <code>dsilog</code> es iniciado el 1 de enero y registra datos para los meses de enero y febrero. Antes de que el archivo <code>dsilog</code> registre los datos de marzo, suprimirá todo el mes de enero.</p>

Control del tamaño del archivo de registro

El usuario determinará cuantos datos serán almacenados en cada clase y cuántos datos serán suprimidos para hacer sitio a nuevos datos.

La capacidad de clase se calcula con INDEX BY (hora, día o mes), RECORDS PER HOUR y MAX INDEXES. Los siguientes ejemplos muestran los resultados de las diversas configuraciones.

En el siguiente ejemplo la capacidad de clase es 288 (24 índices * 12 registros por hora).

```
INDEX BY HOUR
MAX INDEXES 24
RECORDS PER HOUR 12
```

En el siguiente ejemplo la capacidad de clase es 504 (7 días * 24 horas por día * 3 registros por hora).

```
INDEX BY DAY
MAX INDEXES 7
RECORDS PER HOUR 3
```

En el siguiente ejemplo la capacidad de clase es 14.880 (2 meses * 31 días por mes * 24 horas por día * 10 registros por hora).

```
INDEX BY MONTH
MAX INDEXES 2
RECORDS PER HOUR 10
```

Si no se especifican valores para INDEX BY, RECORDS PER HOUR y MAX INDEXES, DSI usará los valores predeterminados de las descripciones de clase. Consulte “Parámetros predeterminados” bajo [Descripción de CLASE](#) con anterioridad en este capítulo.

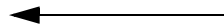
La opción ROLL BY permite determinar cuántos datos suprimir cada vez que se alcance la capacidad de registro de clase. El parámetro ROLL BY está limitado por el parámetro INDEX BY ya que la unidad de ROLL BY (hora, día, mes) no puede ser inferior a la unidad de INDEX BY.

El siguiente ejemplo ofrece una muestra de cómo puede producirse la supresión:

```
INDEX BY DAY
MAX INDEXES 6
ROLL BY DAY
```

Registro de ejemplo
Día 2 - 21 registros
Día 3 - 24 registros
Día 4 - 21 registros
Día 5 - 24 registros
Día 6 - 21 registros

Se liberará espacio cuando la recopilación de datos alcance los 6 días.



El día 7 DSI suprimirá los datos correspondientes al día "más antiguo" haciendo espacio para los registros de datos del día 7.

En el ejemplo anterior la capacidad de clase está limitada a 6 días de datos por el parámetro:

```
MAX INDEXES 6.
```

La supresión de datos está establecida para un día de datos por el parámetro:

```
ROLL BY DAY.
```

Cuando lleguen los datos correspondientes al día 7, el día de datos "más antiguo" será suprimido. Adviértase que al comienzo del proceso de registro no se suprime ningún dato. Una vez que la clase se llene por primera vez al alcanzarse los 7 días, la supresión tendrá lugar una vez al día.

RECORDS PER HOUR

El parámetro `RECORDS PER HOUR` determina cuántos registros serán escritos en el archivo de registro por hora. El número predeterminado de `RECORDS PER HOUR` es 12 para coincidir con el intervalo de medida de muestreo de datos de Componente Performance Collection una vez cada cinco minutos (60 minutos/12 registros = registro cada cinco minutos).

El número predeterminado o el número introducido por el usuario puede requerir el proceso de registro para resumir datos antes de que formen parte del archivo de registro. El método usado para resumir los ítem de datos está especificado en la descripción de métrica. Para mayor información consulte la sección [Método de resumen](#) más adelante en este capítulo.

Sintaxis

```
[RECORDS PER HOUR number]
```

Instrucciones sobre su uso

El proceso de registro usa este valor para resumir datos de entrada con objeto de producir el número de registros especificado. Por ejemplo, si los datos llegan cada minuto y se ha establecido `RECORDS PER HOUR` en 6 (cada 10 minutos), 10 puntos de datos serán resumidos para escribir cada registro en la clase. A continuación se muestran diversas configuraciones habituales de `RECORDS PER HOUR`:

```
RECORDS PER HOUR 6 --> 1 record/10 minutes
RECORDS PER HOUR 12 --> 1 record/5 minutes
RECORDS PER HOUR 60 --> 1 record/minute
RECORDS PER HOUR 120 --> 1 record/30 seconds
```

Advertencia:

RECORDS PER HOUR puede ser invalidado por la opción de segundos `-s seconds` en `dsilog`. No obstante, invalidar la configuración original puede causar problemas al desplegar los datos en gráficas Performance Manager.

Si `dsilog` no recibe datos de métrica durante todo un intervalo de registro, se registrará un indicador de ausencia de datos para esa métrica en particular. Puede forzarse a DSI para que use el último valor registrado con la opción `-asyn` en `dsilog`. Para obtener una descripción de la opción `-asyn`, consulte [Proceso de registro dsilog](#) en Capítulo 15.

Ejemplo

En el siguiente ejemplo se escribirá un registro cada 10 minutos.

```
CLASS VMSTAT_STATS = 10001
LABEL "VMSTAT data"
RECORDS PER HOUR 6;
```

CAPACITY

CAPACITY es el número de registros que serán almacenados en la clase.

Sintaxis

```
[CAPACITY {maximum_record_number}]
```

Instrucciones sobre su uso

La capacidad de clase se deriva de la configuración de RECORDS PER HOUR, INDEX BY y MAX INDEXES. El parámetro CAPACITY es ignorado a menos que se especifique una capacidad mayor que los valores derivados de estas otras configuraciones. Si ocurre una situación como la anteriormente descrita, el parámetro MAX INDEXES será incrementado para proporcionar la capacidad especificada.

Ejemplo

```
INDEX BY DAY
MAX INDEXES 9
RECORDS PER HOUR 12
CAPACITY 3000
```

En el ejemplo anterior la capacidad de clase derivada es 2.592 registros (9 días * 24 horas por día * 12 registros por hora).

Como 3000 es mayor que 2592, `sdlcomp` incrementará MAX INDEXES en 11, lo que tendrá como resultado una capacidad de clase de 3168. Tras la compilación podrán verse los valores resultantes de MAX INDEXES y CAPACITY ejecutando `sdlutil` con la opción `-decomp`.

Descripciones de métrica

Las descripciones de métrica del archivo de especificación de clase permiten definir los ítems de datos de la clase. La descripción de métrica equipara un nombre de métrica con un identificador numérico y especifica el método que será usado cuando los datos tengan que ser resumidos porque estén llegando más registros por hora de los especificados por el usuario con el parámetro `RECORDS PER HOUR`.

- ▶ Las descripciones definidas por el usuario, como por ejemplo *metric_label_name*, no pueden coincidir con ningún elemento de palabra clave de la sintaxis de especificación de clase DSI.

Adviértase que se ha establecido un límite de 100 métricas en el archivo de formato `dsilog`.

```
METRICS

metric_name = metric_id_number
[ LABEL "metric_label_name" ]
[ TOTALED | AVERAGED | SUMMARIZED BY metric_name ]
[ MAXIMUM metric_maximum_number ]
[PRECISION {0 | 1 | 2 | 3 | 4 | 5} ]
TYPE TEXT LENGTH "length"
```

- ▶ En el caso de métrica numérica el usuario podrá especificar el método de resumen (`TOTALED`, `AVERAGED`, `SUMMARIZED BY`) y `PRECISION`. En el caso de métrica de texto el usuario sólo podrá especificar el parámetro `TYPE TEXT LENGTH`.

MÉTRICA

El nombre de métrica y el número de id identifican la métrica que está siendo recopilada.

Sintaxis

```
METRICS
metric_name = metric_id_number
```

Instrucciones sobre su uso

La sección de métrica deberá comenzar con la palabra clave `METRICS` antes de la primera definición de métrica. Todas las métricas deberán tener un nombre de métrica que reúna los requisitos siguientes:

- No deberán contener más de 20 caracteres.
- Deberán comenzar por un carácter alfabético.
- Sólo podrán contener guiones bajos y caracteres alfanuméricos.
- No distinguen entre mayúscula y minúscula

La métrica también tiene un número de ID de métrica cuyo límite es de 6 caracteres.

Los parámetros *metric_name* y *metric_id_number* deberán ser únicos con respecto a toda la métrica definida en la clase. La combinación *class_name:metric_name* deberá ser única para el sistema y no podrá ser igual que ninguna *application_name:metric_name*.

Cada descripción de métrica está separada de la siguiente por un punto y coma (;).

Los nombres de métrica pueden ser reutilizados de otra clase cuyos datos estén almacenados en el mismo conjunto de archivos de registro si las definiciones son también idénticas (consulte [Modo en que están organizados los archivos de registro](#) en Capítulo 13). Para reusar una definición de métrica que haya sido anteriormente definida en otra clase del mismo conjunto de archivos de registro, especifique sólo el parámetro *metric_name* sin *metric_id_number* u otra especificación cualquiera. Si alguna de las opciones es establecida de manera distinta a la métrica anteriormente definida, la métrica deberá tener un nombre e identificador numérico exclusivos y ser redefinida.

El orden de los nombres de métrica en esta sección de la especificación de clase determina el orden de los campos al exportar los datos registrados. Si el orden de los datos de entrada es diferente al orden determinado en esta especificación o si no desean registrar todos los datos en el flujo de datos entrante, consulte [Capítulo 15, Referencia del programa DSI](#) para mayor información sobre cómo asignar la métrica en la ubicación correcta.

Una métrica de marca de hora será insertada automáticamente como la primera métrica de cada clase. Si desea que la marca de hora aparezca en una ubicación distinta en los datos exportados, incluya la forma corta de la definición de métrica definida internamente (DATE_TIME;) en la posición que desee que aparezca. Para omitir la marca de hora y usar una marca de hora UNIX (segundos a partir de 1/1/70 00:00:00) que sea parte de los datos de entrada, seleccione la opción `-timestamp` al comenzar el proceso `dsilog`.

La descripción de métrica más simple, la cual usa el nombre de métrica como etiqueta y los valores predeterminados de las posiciones decimales AVERAGED, MAXIMUM 100 y PRECISION 3, requieren la descripción siguiente:

```
METRICS
metric_name = metric_id_number
```



Deberá compilarse cada clase con `sdlcomp` y a continuación registrar los datos de esa clase con el proceso `dsilog`, independientemente de si se han reusado o no nombres de métrica.

Ejemplo

```
VM;
```

VM es un ejemplo de reutilización de una definición de métrica que ha sido previamente definida en otra clase del mismo conjunto de archivos de registro.

ETIQUETA

La etiqueta de métrica identifica la métrica en datos exportados y gráficas de Performance Manager.

Sintaxis

```
[LABEL "metric_label_name"]
```

Instrucciones sobre su uso

Especifique una cadena de texto entre comillas dobles para etiquetar la métrica en gráficas y datos exportados. Se permiten hasta 48 caracteres. Si no se especifica ninguna métrica, se utilizará el nombre de métrica para identificar la métrica.

Advertencias:

Si la etiqueta contiene dobles comillas, anteponga una barra diagonal inversa (\). Por ejemplo, debería introducir "`\my\ data`" si la etiqueta fuese "my" data.

metric_label_name no podrá coincidir con ninguno de los elementos de palabra clave de la sintaxis de especificación de clase DSI, como por ejemplo CAPACITY o ACTION, entre otros.

Ejemplo

```
METRICS
RUN_Q_PROCS = 106
LABEL "Procs in run q";
```

Método de resumen

El método de resumen determina cómo resumir datos si el número de registros excede el número establecido en la opción RECORDS PER HOUR de la sección CLASS. Por ejemplo, puede ocurrir que desee obtener el recuento total de instancias y a la vez el promedio de una frecuencia. El método de resumen sólo es válido con métricas numéricas.

Sintaxis

```
[{TOTALED | AVERAGED | SUMMARIZED BY metric_name}]
```

Instrucciones sobre su uso

Deberá usarse SUMMARIZED BY cuando una métrica no sea promediada en el tiempo sino con respecto a otra métrica de la clase. Por ejemplo, imagine que ha definido las métricas TOTAL_ORDERS y LINES_PER_ORDER. Si estas métricas entran en el proceso de registro cada cinco minutos pero los registros sólo se escriben por hora, para resumir correctamente LINES_PER_ORDER (total de líneas / total de pedidos), el proceso de registro deberá efectuar el siguiente cálculo cada cinco minutos:

- Multiplicar LINES_PER_ORDER * TOTAL_ORDERS al final de cada intervalo de 5 minutos y mantener el resultado en un recuento en ejecución interno de total de líneas.
- Mantener el recuento en ejecución de TOTAL_ORDERS.
- Al final de la hora, dividir el total de líneas por TOTAL_ORDERS.

Para especificar este tipo de cálculo, especifique LINES_PER_ORDER como SUMMARIZED BY TOTAL_ORDERS.

Si no se ha especificado ningún método de resumen, la métrica utilizará como valor predeterminado AVERAGED.

Ejemplo

```
METRICS
ITEM_1_3 = 11203
LABEL "TOTAL_ORDERS"
TOTALED;
ITEM_1_5 = 11205
LABEL "LINES_PER_ORDER"
SUMMARIZED BY ITEM_1_3;
```

PRECISION

PRECISION identifica el número de posiciones decimales que será usado con los valores de métrica. Si PRECISION no es especificado, éste será calculado según el MAXIMUM especificado. Si no se especifica ninguno de los dos, el valor predeterminado de PRECISION será 3. Esta configuración es sólo válida con métricas numéricas.

Sintaxis

```
[PRECISION{0|1|2|3|4|5}]
```

Instrucciones sobre su uso

El parámetro PRECISION determina el valor más alto que puede ser registrado. Use PRECISION 0 con números enteros.

PRECISION	Núm. posic. decimales	Número más alto aceptable	MAXIMUM
0	0	2,147,483,647	> 10,000
1	1	214,748,364.7	de 1.001 a 10.000
2	2	21,474,836.47	de 101 a 1.000
3	3	2,147,483.647	de 11 a 1.000
4	4	214,748.3647	de 2 a 10
5	5	21,474.83647	1

Ejemplo

```
METRICS  
RUN_Q_PROCS      = 106  
LABEL           "Procs in run q"  
PRECISION 1;
```

TYPE TEXT LENGTH

Las tres palabras clave TYPE TEXT LENGTH especifican que la métrica es de texto en lugar de numérica. El texto se define como cualquier caracter distinto de ^d, \n, o el separador, si lo hubiera.

Dado que el delimitador predeterminado entre los ítems de datos de dsilog es un espacio en blanco, será preciso cambiar el delimitador si el texto contiene espacios insertados. Use la opción dsilog -c char para especificar un separador distinto como se describe en el [Capítulo 15, Referencia del programa DSI](#).

Sintaxis

```
[TYPE TEXT LENGTH length]
```

Instrucciones sobre su uso

length deberá ser superior a cero e inferior a 4.096.

Advertencia:

El método de resumen, MAXIMUM y PRECISION no podrán ser especificados con métrica de texto. El texto no puede ser resumido, lo que significa que dsilog tomará el primer valor registrado en un intervalo e ignorará el resto.

Ejemplo

```
METRICS
text_1 = 16
LABEL "first text metric"
TYPE TEXT LENGTH 20
;
```

Especificación de clase de ejemplo

```
CLASS VMSTAT_STATS = 10001
LABEL "VMSTAT data"
INDEX BY HOUR
MAX INDEXES 12
ROLL BY HOUR
RECORDS PER HOUR 120;

METRICS

RUN_Q_PROCS = 106
LABEL "Procs in run q"
PRECISION 0;

BLOCKED_PROCS = 107
LABEL "Blocked Processes"
PRECISION 0;

SWAPPED_PROCS = 108
LABEL "Swapped Processes"
PRECISION 0;

AVG_VIRT_PAGES = 201
LABEL "Avg Virt Mem Pages"
PRECISION 0;

FREE_LIST_SIZE = 202
LABEL "Mem Free List Size"
PRECISION 0;

PAGE_RECLAIMS = 303
LABEL "Page Reclaims"
PRECISION 0;

ADDR_TRANS_FAULTS = 304
LABEL "Addr Trans Faults"
PRECISION 0;
```

```
PAGES_PAGED_IN      = 305
LABEL               "Pages Paged In"
PRECISION 0;
```

```
PAGES_PAGED_OUT    = 306
LABEL               "Pages Paged Out"
PRECISION 0;
```

```
PAGES_FREED        = 307
LABEL               "Pages Freed/Sec"
PRECISION 0;
```

```
MEM_SHORTFALL      = 308
LABEL               "Exp Mem Shortfall"
PRECISION 0;
```

```
CLOCKED_PAGES      = 309
LABEL               "Pages Scanned/Sec"
PRECISION 0;
```

```
DEVICE_INTERRUPTS = 401
LABEL               "Device Interrupts"
PRECISION 0;
```

```
SYSTEM_CALLS       = 402
LABEL               "System Calls"
PRECISION 0;
```

```
CONTEXT_SWITCHES  = 403
LABEL               "Context Switches/Sec"
PRECISION 0;
```

```
USER_CPU           = 501
LABEL               "User CPU"
PRECISION 0;
```

```
SYSTEM_CPU         = 502
LABEL               "System CPU"
PRECISION 0;
```

```
IDLE_CPU           = 503
LABEL               "Idle CPU"
PRECISION 0;
```

15 Referencia del programa DSI

El presente capítulo ofrece información de referencia detallada sobre:

- el compilador `sdlcomp`
- los archivos de configuración `datasources` y `alarmdef`
- el proceso de registro `dsilog`
- exportar datos DSI con el programa `extract` de Componente Performance Collection
- la utilidad de administración de orígenes de datos `sdlutil`

Compilador `sdlcomp`

El compilador `sdlcomp` comprueba el archivo de especificación de clase por si se han producido errores. Si no se detecta ningún error, agrega las descripciones de clase y métrica al archivo de descripción en el conjunto de archivos de registro designado por el usuario. Crea también los señaladores en los archivos de raíz del conjunto de archivos de registro que serán usados para almacenamiento de datos. Si no existe o bien el conjunto de archivos de registro o el archivo de registro, éste será creado por el compilador.



Los archivos DSI podrán ponerse en cualquier lugar del sistema especificando la ruta completa en el comando del compilador. No obstante, una vez que se ha especificado la ruta, los archivos de registro DSI *no podrán* ser cambiados a otros directorios. (SDL62 es el mensaje de error de especificación de clase asociado descrito en [Mensajes de error SDL](#) en Capítulo 17. El formato usado por DSI para los mensajes de error de especificación de clase es el prefijo `SDL` (archivo de registro autodescriptivo o 'Self Describing Logfile'), seguido del número de mensaje.

Sintaxis del compilador

```
sdlcomp [-maxclass value] specification_file  
[logfile_set[log file]] [options]
```

Variables y opciones	Definiciones
-maxclass <i>value</i>	permite especificar el número de clases máximo que será proporcionado para crear un nuevo conjunto de archivos de registro. Esta opción es ignorada si se usa con el nombre de un conjunto de archivos de registro existente. Cada clase adicional consume unos 500 bytes de sobrecarga de espacio de disco, tanto si se usa la clase como si no. El valor predeterminado es 10 si no se especifica -maxclass.
<i>specification_file</i>	es el nombre del archivo que contiene la especificación de clase. Si no está en el directorio actual, deberá aparecer el nombre completo.
logfile_set	es el nombre del conjunto de archivos de registro a la que debería ser agregada esta clase.
log file	es el archivo de registro del conjunto que contiene los datos correspondientes a esta clase. Si no se designa ningún archivo de registro, se creará un nuevo archivo de registro para la clase y se le asignará un nombre automáticamente.
-verbose	imprime una descripción detallada de la salida del compilador en <code>stdout</code> .
-vers	muestra la información de versión.
-?	muestra la descripción de sintaxis.
-u	permite registrar más de un registro por segundo. Use esta opción sólo para registrar datos no resumidos.

Salida de compilador de ejemplo

En la línea de comando siguiente:

```
sdlcomp vmstat.spec sdl_new
```

el siguiente código contiene la salida de ejemplo de una compilación correcta. Adviértase que `vmstat.spec` es el archivo de especificación de ejemplo descrito en el capítulo anterior.

```
sdlcomp  
Check class specification syntax.
```

```
CLASS VMSTAT_STATS = 10001  
LABEL "VMSTAT data"  
INDEX BY HOUR  
MAX INDEXES 12  
ROLL BY HOUR  
RECORDS PER HOUR 120;  
  
METRICS  
  
RUN_Q_PROCS = 106  
LABEL "Procs in run q"  
PRECISION 0;  
  
BLOCKED_PROCS = 107  
LABEL "Blocked Processes"  
PRECISION 0;  
  
SWAPPED_PROCS = 108  
LABEL "Swapped Processes"  
PRECISION 0;  
  
AVG_VIRT_PAGES = 201  
LABEL "Avg Virt Mem Pages"  
PRECISION 0;  
  
FREE_LIST_SIZE = 202  
LABEL "Mem Free List Size"  
PRECISION 0;  
  
PAGE_RECLAIMS = 303  
LABEL "Page Reclaims"  
PRECISION 0;  
ADDR_TRANS_FAULTS = 304  
LABEL "Addr Trans Faults"  
PRECISION 0;  
  
PAGES_PAGED_IN = 305  
LABEL "Pages Paged In"  
PRECISION 0;  
  
PAGES_PAGED_OUT = 306  
LABEL "Pages Paged Out"  
PRECISION 0;
```

```

PAGES_FREED          = 307
LABEL                "Pages Freed/Sec"
PRECISION 0;

MEM_SHORTFALL        = 308
LABEL                "Exp Mem Shortfall"
PRECISION 0;

CLOCKED_PAGES        = 309
LABEL                "Pages Scanned/Sec"
PRECISION 0;

DEVICE_INTERRUPTS    = 401
LABEL                "Device Interrupts"
PRECISION 0;

SYSTEM_CALLS         = 402
LABEL                "System Calls"
PRECISION 0;

CONTEXT_SWITCHES     = 403
LABEL                "Context Switches/Sec"
PRECISION 0;

USER_CPU              = 501
LABEL                "User CPU"
PRECISION 0;

SYSTEM_CPU            = 502
LABEL                "System CPU"
PRECISION 0;

IDLE_CPU              = 503
LABEL                "Idle CPU"
PRECISION 0;

```

Nota: Time stamp inserted as first metric by default.

Syntax check successful.

```

Update SDL sdl_new.
Open SDL sdl_new
Add class VMSTAT_STATS.
Check class VMSTAT_STATS.

```

Class VMSTAT_STATS successfully added to log file set.

Para mayor información sobre mensajes de error y recuperación, consulte el [Capítulo 17, Mensajes de error](#).

Archivos de configuración

Antes de comenzar a registrar datos puede ser que necesite actualizar dos archivos de configuración de Componente Performance Collection:

- `/var/opt/OV/conf/perf/datasources`
- `/var/opt/perf/alarmdef` — consulte la sección siguiente, [Definición de alarmas en métrica DSI](#) para información sobre el uso del archivo de configuración `alarmdef`.

Definición de alarmas en métrica DSI

Componente Performance Collection permite definir alarmas en métrica DSI. Estas alarmas notifican al usuario si la métrica DSI excede las condiciones establecidas. Para definir alarmas el usuario deberá especificar las condiciones que activen una acción o notificación de alerta cuando dichas condiciones se cumplan o se excedan. Las alarmas de datos registrados en DSI se definen del mismo modo que para otras métricas de Componente Performance Collection (en el archivo `alarmdef` del sistema de Componente Performance Collection). El archivo `alarmdef` se encuentra en el directorio de configuración `var/opt/perf/` de Componente Performance Collection.

Cuando el usuario especifique un nombre de métrica DSI en una definición de alarma, deberá ser un nombre completo; esto es, precedido por el `datasource_name` y el `class_name` como se muestra a continuación:

```
datasource_name:class_name:metric_name
```

- `datasource_name` es el nombre usado para configurar el origen de datos en el archivo `datasources`.
- `class_name` es el nombre usado para identificar la clase en la especificación de clase correspondiente al origen de datos. No se necesita introducir el `class_name` si el nombre de métrica es único (no reutilizado) en la especificación de clase.
- `metric_name` es el ítem de datos de la especificación de clase correspondiente al origen de datos.

No obstante, si no selecciona un nombre de métrica completo, necesitará incluir la instrucción `USE` en el archivo `alarmdef` para identificar el origen de datos que será usado. Para más información sobre la instrucción `USE`, consulte el Capítulo 7 “Alarmas de rendimiento,” del Manual del usuario UNIX de *HP Operations Agent*.

Para activar los cambios realizados en el archivo `alarmdef` de manera que pueda ser leído por el generador de alarmas, introduzca el comando **ovpa restart alarm** en la línea de comandos.

Para información detallada sobre la sintaxis de definiciones de alarma, cómo son procesadas las alarmas y personalizar definiciones de alarma, consulte el Capítulo 7 del Manual del usuario Unix de *HP Operations Agent*.

Procesamiento de alarmas

A medida que `dsilog` registra datos, éstos son comparados con las definiciones de alarma del archivo `alarmdef` para determinar si una condición se cumple o se excede. Cuando ello ocurra, se activará una acción o notificación de alerta.

Se pueden configurar dónde se desean notificaciones de alarma enviadas y si se desean acciones locales realizadas. Las notificaciones de alarma pueden ser enviadas al sistema de análisis central de Performance Manager donde se podrán crear las gráficas de métrica que caracterizan el rendimiento del sistema. Las capturas SNMP pueden ser enviadas al Network Node Manager de HP. Las acciones locales pueden ser realizadas en el sistema de Componente Performance Collection. La información de alarma puede ser enviada también a Operations Manager.

Proceso de registro dsilog

El proceso `dsilog` requiere que bien el usuario cree su propio programa o que uso uno ya existente para ganar acceso a los datos. El usuario podrá entonces canalizar estos datos a `dsilog`, el cual registrará los datos en el conjunto de archivos de registro. Se deberá utilizar un proceso de registro independiente por cada clase definida.

`dsilog` espera recibir datos de `stdin`. Para iniciar el proceso de registro, canalice los resultados del proceso que esté utilizando para recopilar datos a `dsilog` como se muestra en el ejemplo siguiente.

```
vmstat 60 | dsilog logfile_set class
```

Sólo se puede tener una barra vertical (`|`) en la línea de comandos. El motivo es que cuando se usan dos barras verticales, el proceso de búfer de UNIX conserva la salida del primer comando hasta que se hayan escrito 8.000 caracteres para proseguir con el segundo comando y canalizar la salida al archivo de registro.

También puede usarse un `fifo` (canalización con nombre). Por ejemplo:

```
mkfifo -m 777 myfifo
dsilog logfile_set class -i myfifo &
vmstat 60 > myfifo &
```

El signo `&` hace que el proceso sea ejecutado de fondo.

Adviértase que puede ser preciso incrementar los valores de los parámetros de núcleo de UNIX `shmmni` y `nfllocks` si se tiene programado ejecutar un alto número de procesos `dsilog`. `Shmmni` especifica el máximo de segmentos de memoria compartida; `nfllocks` especifica el máximo de bloqueos de archivos en el sistema. El valor predeterminado de cada uno es 200. Cada archivo de registro DSI activo usa un segmento de memoria compartida (`shmmni`) y uno o más bloqueos de archivos (`nfllocks`). En HP-UX se puede cambiar la configuración de `shmmni` y `nfllocks` con la utilidad System Administration and Maintenance (SAM).

Sintaxis

```
dsilog logfile_set class [options]
```

Los parámetros y opciones de `dsilog` se describen en las páginas siguientes.

Tabla 1 parámetros y opciones de `dsilog`

Variables y opciones	Definiciones
<code>logfile_set</code>	es el nombre del archivo de registro establecido donde se almacenarán los datos. Si no está en el directorio actual, deberá aparecer el nombre completo.
<code>class</code>	es el nombre de la clase que se va a registrar.
<code>-asyn</code>	indica que los datos llegarán de forma asíncrona con la frecuencia <code>RECORDS PER HOUR</code> . Si no llegan datos durante un intervalo de registro, se repetirán los datos del último intervalo de registro. No obstante, si <code>dsilog</code> no ha registrado aún datos, el valor de métrica registrado será tratado como datos ausentes. Esto hace que se dibuje una línea plana en una pantalla gráfica de los datos y que éstos se repitan en cada registro al exportarlos.
<code>-c char</code>	usa el caracter especificado como separador/delimitador de cadena. No se pueden utilizar los elementos siguientes como separadores: decimal, signo menos, <code>^z</code> , <code>\n</code> . Si hay espacios insertados en la métrica de texto, entonces deberá especificar un separador único con esta opción.
<i>archivo de formato</i> <code>-f</code>	<p>especifica un archivo que describe los datos que se introducirán en el proceso de registro. Si no se especifica esta opción, <code>dsilog</code> derivará el formato de la entrada de la especificación de clase con las suposiciones siguientes. Para mayor información consulte la sección Creación de un archivo de formato más adelante en este capítulo.</p> <p>Cada ítem de datos de un registro de entrada corresponde a una métrica definida en la especificación de clase.</p> <p>Las métricas se definen en la especificación de clase en el orden en el que aparecen como ítems de datos en el registro de entrada.</p> <p>Si hay más ítems de datos en un registro de entrada que definiciones de métrica, <code>dsilog</code> ignorará todos los ítems de datos adicionales.</p>
<i>archivo de formato</i> <code>-f</code> (continuación)	<p>Si la especificación de clase enumera más definiciones de métrica que elementos de datos de entrada existentes, el campo mostrará los datos “ausentes” cuando se exporten los datos y no habrá datos disponibles para esa métrica al crear gráficos con los datos en el software de análisis.</p> <p>El número de campos en el archivo de formato ha sido limitado en 100.</p>

Tabla 1 parámetros y opciones de `dsilog`

Variables y opciones	Definiciones
<code>-i fifo</code> o archivo ASCII	indica que la entrada debe venir del <code>fifo</code> o archivo ASCII con nombre. Si no se utiliza esta opción, la entrada procederá de <code>stdin</code> . Si se utiliza este método, inicie <code>dsilog</code> antes de iniciar el proceso de recopilación. Consulte la página <code>mkfifo</code> para obtener más información sobre el uso de un <code>fifo</code> . Consulte el Capítulo 16, Ejemplos de integración de orígenes de datos (DSI) para obtener ejemplos.
<code>-s seconds</code>	es el número de segundos en el que se resumen los datos. La opción <code>-s</code> invalida el intervalo de resumen y la frecuencia de resumen se establece en el valor predeterminado <code>RECORDS PER HOUR</code> en la especificación de clase. Si se encuentra presente, esta opción invalida el valor de <code>RECORDS PER HOUR</code> . Un cero (0) desactiva el resumen, lo que significa que se registrarán todos los datos entrantes. Deberá tenerse precaución al usar la opción <code>-s 0</code> porque <code>dsilog</code> agregará una marca de tiempo en los datos de registro en el momento en que se recibe el punto. Ello puede causar problemas en Performance Manager y <code>perfalarm</code> , los cuales funcionan de manera más óptima con marcas de tiempo en intervalos regulares. Si el archivo de registro va a ser accedido por Performance Manager, se desaconseja usar la opción <code>-s 0</code> .
<code>-t</code>	imprime todo lo que está registrado en <code>stdout</code> en formato ASCII.
<code>-timestamp</code>	indica que el proceso de registro no debe ofrecer la marca de hora, sino utilizar la que se proporciona en los datos de entrada. La marca de hora en los datos entrantes debe tener un formato de marca de hora UNIX (segundos desde 1/1/70 00:00:00) y representar la hora local.
<code>-vi</code>	filtra la entrada a través de <code>dsilog</code> y escribe los errores en <code>stdout</code> en lugar de en el archivo de registro. No escribe los datos registrados actuales en <code>stdout</code> (consulte la opción <code>-vo</code> descrita a continuación). Esta opción se puede utilizar para comprobar la validez de la entrada.
<code>-vo</code>	filtra la entrada a través de <code>dsilog</code> y escribe los los errores y datos registrados actuales en <code>stdout</code> en lugar de en el archivo de registro. Esta opción se puede utilizar para comprobar la validez del resumen de datos.
<code>-vers</code>	muestra la información de versión.

Tabla 1 parámetros y opciones de dsilog

Variables y opciones	Definiciones
-?	muestra la descripción de sintaxis.

Procesamiento de datos de dsilog

El programa `dsilog` explora cada cadena de datos de entrada, analizando los campos delimitados en métrica de texto o métrica numérica independiente. Una regla clave para predecir el modo de procesamiento de los datos es la validez de la cadena de entrada. Una cadena de entrada válida requiere que esté presente un delimitador entre los tipos de métrica especificados (numéricos o de texto). Un espacio en blanco es un delimitador predeterminado, pero se puede especificar un delimitador diferente con la opción de línea de comando `dsilog -c char`.

Deberá incluirse un caracter de línea nuevo al final del registro ingresado en DSI con objeto de que DSI lo interprete correctamente.

Pruebas del proceso de registro con Sdlgendata

Antes de comenzar a registrar datos, puede probar el conjunto de archivos de registro compilado y el proceso de registro usando el programa `sdlgendata`. `sdlgendata` identifica la métrica de una clase (como se describe en la especificación de clase) y genera datos para cada métrica de una clase.

Sintaxis

```
sdlgendata logfile_set class [options]
```

A continuación se ofrece una explicación sobre los parámetros y opciones de `Sdlgendata`.

Tabla 2 parámetros y opciones de Sdlgendata

Variables y opciones	Definiciones
<code>logfile_set</code>	es el nombre del conjunto de archivos de registro para el que generar datos.
<code>class</code>	es la clase de datos para la que generar datos.
<code>-timestamp [number]</code>	ofrece una marca de tiempo con los datos. Si se ofrece un número negativo o no se ofrece ningún número en absoluto, se usará la hora actual para <code>timestamp</code> . Si se usa un número positivo, la hora comenzará en 0 y será incrementado por <i>número</i> para cada registro de datos nuevo.
<code>-wait number</code>	ocasiona una espera de <i>número</i> segundos entre los registros generados.

Tabla 2 parámetros y opciones de Sdlgendata

Variables y opciones	Definiciones
-cycle number	recicla datos transcurridos <i>número</i> ciclos.
-vers	muestra la información de versión.
-?	muestra la descripción de sintaxis.

Al canalizar la salida de `sdlgendata` en `dsilog` con bien la opción `-vi` o `-vo`, podrá verificar la entrada (`-vi`) y la salida (`-vo`) antes de empezar a registrar con su propio proceso o programa.



Una vez que haya finalizado las pruebas, elimine de la prueba los archivos de registro creados. De lo contrario, estos archivos seguirán siendo parte de la prueba de archivo de registro.

Use el comando siguiente para canalizar datos de `sdlgendata` al proceso de registro. La opción `-vi` especifica que los datos son eliminados y los errores escritos en `stdout`. Pulse **CTRL+C** u otro caracter de control de interrupción para detener la generación de datos.

```
sdlgendata logfile_set class -wait 5 | dsilog \  
logfile_set class -s 10 -vi
```

El comando anterior genera datos cuyo aspecto es el siguiente:

```
dsilog  
I: 744996402 1.0000 2.0000 3.0000 4.0000 5.0000 6.0000 7.0000  
I: 744996407 2.0000 3.0000 4.0000 5.0000 6.0000 7.0000 8.0000  
I: 744996412 3.0000 4.0000 5.0000 6.0000 7.0000 8.0000 9.0000  
I: 744996417 4.0000 5.0000 6.0000 7.0000 8.0000 9.0000 10.0000  
I: 744996422 5.0000 6.0000 7.0000 8.0000 9.0000 10.0000 11.0000  
I: 744996427 6.0000 7.0000 8.0000 9.0000 10.0000 11.0000 12.0000  
I: 744996432 7.0000 8.0000 9.0000 10.0000 11.0000 12.0000 13.0000  
I: 744996437 8.0000 9.0000 10.0000 11.0000 12.0000 13.0000 14.0000
```

También podrá usar la opción `-vo` de `dsilog` para examinar entradas y resumir salidas de sus propios datos sin necesidad de registrarlos. El siguiente comando canaliza `vmstat` en intervalos de 5 segundos en `dsilog`, donde es resumido a 10 segundos.

```
->vmstat 5 | dsilog logfile_set class -s 10 -vo
dsilog
I: 744997230 0.0000 0.0000 21.0000 2158.0000 1603.0000 2.0000 2.0000
I: 744997235 0.0000 0.0000 24.0000 2341.0000 1514.0000 0.0000 0.0000
interval marker
L: 744997230 0.0000 0.0000 22.5000 2249.5000 1558.5000 1.0000 1.0000

I: 744997240 0.0000 0.0000 23.0000 2330.0000 1513.0000 0.0000 0.0000
I: 744997245 0.0000 0.0000 20.0000 2326.0000 1513.0000 0.0000 0.0000
interval marker
L: 744997240 0.0000 0.0000 21.5000 2328.0000 1513.0000 0.0000 0.0000

I: 744997250 0.0000 0.0000 22.0000 2326.0000 1513.0000 0.0000 0.0000
I: 744997255 0.0000 0.0000 22.0000 2303.0000 1513.0000 0.0000 0.0000
interval marker
L: 744997250 0.0000 0.0000 22.0000 2314.5000 1513.0000 0.0000 0.0000

I: 744997260 0.0000 0.0000 22.0000 2303.0000 1512.0000 0.0000 0.0000
I: 744997265 0.0000 0.0000 28.0000 2917.0000 1089.0000 9.0000 33.0000
interval marker
L: 744997260 0.0000 0.0000 25.0000 2610.0000 1300.5000 4.5000 16.5000

I: 744997270 0.0000 0.0000 28.0000 2887.0000 1011.0000 3.0000 9.0000
I: 744997275 0.0000 0.0000 27.0000 3128.0000 763.0000 8.0000 6.0000
interval marker
L: 744997270 0.0000 0.0000 27.5000 3007.5000 887.0000 5.5000 12.5000
```

También puede usar la opción `-vo` de `dsilog` si desea utilizar un archivo de datos antiguos para pruebas siempre y cuando los datos contengan su propia marca de tiempo UNIX (segundos desde 1/1/70 00:00:00). Para usar un archivo de datos antiguos, introduzca un comando como el siguiente:

```
dsilog -timestamp -vo <oldfile>
```

Creación de un archivo de formato

Cree un archivo de formato para asignar la entrada de datos a la especificación de clase si:

- la entrada de datos contiene datos que no están incluidos en la especificación de clase.
- los datos de entrada contienen métrica en un orden distinto al especificado por el usuario en la especificación de clase.

Un archivo de formato es un archivo de texto ASCII que el usuario puede crear con `vi` u otro editor de texto cualquiera. Use la opción `-f` de `dsilog` para especificar el nombre completo del archivo de formato.

Dado que el proceso de registro funciona realizando una búsqueda del primer carácter válido existente después de un delimitador (bien un espacio por defecto o uno definido por el usuario con la opción `-c` de `dsilog`) para empezar la siguiente métrica, el archivo de formato indica al proceso de registro qué campos omitir y qué nombres de métrica asociar con los campos no omitidos.

`$numeric` indica al proceso de registro que omita un campo de métrica numérica y siga en el siguiente. `$any` indica al proceso de registro que omita un campo de métrica de texto y siga en el siguiente. Adviértase que el archivo de formato está delimitado en 100 campos.

Por ejemplo, si el flujo de datos de entrada contiene la siguiente información:

```
ABC 987 654 123 456
```

y desea registrar sólo el primer campo numérico en una métrica llamada `metric_1`, el archivo de formato presentará el aspecto siguiente:

```
$any metric_1
```

Esto indica al proceso de registro que registre sólo la información contenida en el primer campo numérico y descarte el resto de los datos. Para registrar sólo la información contenida en el tercer campo numérico, el archivo de formato presentará el aspecto siguiente:

```
$any $numeric $numeric metric_1
```

Para registrar los cuatro ítems de datos numéricos, en el orden inverso, el archivo de formato presentará el aspecto siguiente:

```
$any metric_4 metric_3 metric_2 metric_1
```

Si el flujo de datos de entrada contiene la información siguiente:

```
/users    15.9    3295    56.79%    xdisk1 /dev/dsk/  
c0d0s*
```

y sólo desea registrar la primera métrica de texto y los dos primeros campos numéricos en campos de métrica que ha llamado *text_1*, *num_1* y *num_2*, respectivamente, el archivo de formato presentará el siguiente aspecto:

```
text_1    num_1    num_2
```

Esto indica al proceso de registro que registre sólo la información contenida en los primeros tres campos y descarte el resto de los datos.

Para registrar todos los datos, pero eliminar el “%” después de la tercera métrica, el archivo de formato presentará el siguiente aspecto:

```
text_1    num_1    num_2    num_3    $any    text_2    text_3
```

Puesto que está registrando campos numéricos y el “%” es considerado un campo de texto, necesitará omitirlo para registrar correctamente el campo de texto que le sigue.

Para registrar los ítems de datos en un orden distinto, el archivo de formato presentará el siguiente aspecto:

```
text_3    num_2    num_1    num_3    $any    text_2    text_1
```

Adviértase que ello sólo tendrá como resultado el registro de los seis primeros caracteres de *text_3* si la extensión de *text_1* es de seis caracteres en la especificación de clase. Para registrar *todo* en *text_3* como el primer valor, cambie la especificación de clase y modifique el flujo de datos para ganar espacio.

Cambio de especificación de clase

Para cambiar un archivo de especificación de clase deberá recrear el conjunto de archivos de registro en su totalidad como sigue:

- 1 Pare el proceso de `dsilog`.
- 2 Exporte los datos del archivo de registro existente con la opción de marca de tiempo UNIX si desea guardar o integrar los datos antiguos con los nuevos datos que va a registrar. Consulte [Exportación de datos DSI](#) más adelante en este capítulo para mayor información sobre cómo realizar esta acción.
- 3 Ejecute `sdlutil` para eliminar el conjunto de archivos de registro. Consulte [Administrar datos con sdlutil](#) más adelante en este capítulo para mayor información sobre cómo realizar esta acción.
- 4 Actualice el archivo de especificación de clase.
- 5 Ejecute `sdlcomp` para volver a compilar la especificación de clase.
- 6 De manera opcional, use la opción `-i` de `dsilog` para integrar los datos antiguos exportados en el paso 2. Puede que sea necesario manipular los datos para que estén en consonancia con los datos nuevos con ayuda de la opción `-f format_file`.
- 7 Ejecute `dsilog` para comenzar el proceso de archivo en base a la nueva especificación de clase.
- 8 Siempre y cuando no haya cambiado el nombre o ubicación del conjunto de archivos de registro, no necesitará actualizar el archivo `datasources`.

Exportación de datos DSI

Para exportar los datos de un archivo de registro DSI, use la función `export` del programa `extract` de Componente Performance Collection. Consulte los capítulos 5 y 6 del *Manual del usuario UNIX de HP Operations Agent* para información sobre cómo usar `extract` para exportar datos. En la página siguiente se muestra un ejemplo de cómo exportar datos DSI con los argumentos de línea de comando.

Existen varios modos de averiguar qué clases y métrica pueden ser exportadas del archivo de registro DSI. Se puede usar `sdlutil` para obtener esta información como se describe en [Administrar datos con sdlutil](#) más adelante en este capítulo. O puede usar el comando `extract guide` para crear un archivo de plantilla de exportación que incluya las clases y métrica en el archivo de registro DSI. A continuación podrá usar `vi` para editar, asignar nombre o guardar el archivo. El archivo de plantilla de exportación permite especificar el formato de exportación, como se describe en los capítulos 5 y 6 del *Manual del usuario UNIX de HP Operations Agent*.



Para exportar datos del archivo de registro DSI *deberá* ser el usuario raíz o el creador del archivo de registro.

Ejemplo de uso del programa `extract` para exportar datos del archivo de registro DSI

```
extract -xp -l logfile_set -C class [options]
```

Las opciones de línea de comando `extract` permiten hacer lo siguiente:

- Especificar un archivo de salida de exportación.
- Establecer las horas y fechas de comienzo y fin del primer y último intervalo a exportar.
- Exportar datos sólo entre determinadas horas (turnos).
- Excluir datos de ciertos días de la semana (como por ejemplo fines de semana).
- Especificar el caracter de separación que se colocará entre la métrica en los informes.
- Seleccionar si se desplegarán encabezados y registros en blanco durante intervalos en los que no lleguen datos y cuál será el valor desplegado en el caso de datos nulos o inexistentes.
- Mostrar en pantalla hora/fecha exportado en formato UNIX o formato de hora y fecha.
- Establecer niveles de resumen adicionales.

Ver datos en Performance Manager

Para mostrar datos de un archivo de registro DSI en Performance Manager, necesitará configurar el archivo de registro DSI como origen de datos de Componente Performance Collection. Antes de comenzar a registrar datos, configure el origen de datos agregándolo al archivo `datasources` del sistema de Componente Performance Collection.

Performance Manager permite ver de manera central, monitorizar, analizar, comparar y extrapolar tendencias. Performance Manager ayuda a identificar problemas actuales y potenciales. Ofrece la información necesaria para resolver problemas antes de que se vea afectada la productividad del usuario.

Administrar datos con sdlutil

Si desea administrar los datos de un archivo de registro DSI, use el programa `sdlutil` para realizar alguna de las siguientes acciones:

- enunciar la información de métrica y la clase definida actuales en `stdout`. Es posible redireccionar la salida a un archivo.
- enunciar estadísticas de clases completas en `stdout`.
- mostrar descripciones de métrica de todas las métricas enumeradas.
- enumerar los archivos en un conjunto de archivos de registro.
- suprimir clases y datos de un conjunto de archivos de registro.
- volver a crear una especificación de clase a partir de la información del conjunto de archivos de registro.
- mostrar información de la versión.

Sintaxis

```
sdlutil logfile_set [option]
```

Variables y opciones	Definiciones
<code>logfile_set</code>	es el nombre de un conjunto de archivos de registro creado mediante la compilación de una especificación de clase.
<code>-classes classlist</code>	proporciona una descripción de clase de todas las clases enumeradas. Si no se enumera ninguna, se proporcionarán todas. Separe los ítems en la <code>classlist</code> con espacios.
<code>-stats classlist</code>	proporciona estadísticas completas de todas las clases enumeradas. Si no se enumera ninguna, se proporcionarán todas. Separe los ítems en la <code>classlist</code> con espacios.
<code>-metrics metriclist</code>	ofrece descripciones de métrica de toda la métrica contenida en la <code>metriclist</code> . si no se enumera ninguna, se proporcionarán todas. Separe los ítems en la <code>metriclist</code> con espacios.
<code>-id</code>	muestra el ID del segmento de memoria compartida que utiliza el archivo de registro.
<code>-files</code>	enumera todos los archivos del conjunto de archivos de registro.
<code>-rm all</code>	suprime todas las clases y datos así como sus datos y el ID de memoria compartida del archivo de registro.

Variables y opciones	Definiciones
-decomp <i>classlist</i>	vuelve a crear una especificación de clase a partir de la información del conjunto de archivos de registro. Los resultados se escriben en <code>stdout</code> y se deben redireccionar a un archivo si se prevé realizar cambios en el archivo y utilizarlo de nuevo. Separe los ítems en la <i>classlist</i> con espacios.
-vers	muestra información de la versión.
-?	muestra la descripción de sintaxis.

16 Ejemplos de integración de orígenes de datos (DSI)

La integración de orígenes de datos es una tecnología muy potente y flexible. La implementación de DSI puede abarcar implementaciones sencillas y fáciles e implementaciones muy complejas.

Este capítulo contiene ejemplos del uso de DSI para las siguientes tareas:

- escribir una secuencia de comandos `dsilog`
- registrar datos `vmstat`
- registrar datos `sar`
- registrar un recuento de palabras `who`

Escritura de una secuencia de comandos dsilog

El código `dsilog` se ha diseñado para recibir como entrada un flujo continuo de filas de datos. Este flujo de entrada se resume en `dsilog`, de acuerdo con las directivas de especificación de cada clase, y se registra una fila de datos resumida por intervalo de resumen solicitado. Performance Manager y `perfalarm` funcionan mejor cuando las marcas de hora escritas en el registro se ajustan a la velocidad de resumen esperada (registros por hora). Esto ocurre automáticamente cuando se permite a `dsilog` llevar a cabo el resumen.

`dsilog` procesa cada fila de entrada que llega, lo que puede causar problemas con Performance Manager y `perfalarm`. No se recomienda este método.

- Secuencia de comandos `dsilog` problemática
- Secuencia de comandos `dsilog` recomendada

Ejemplo 1 - Secuencia de comandos dsilog problemática

En la secuencia de comandos siguiente, se ejecuta un nuevo proceso `dsilog` para cada fila de entrada que llega.

```
while :
do
    feed_one_data_row | dsilog sdlname classname
    sleep 50
done
```

Ejemplo 2 - Secuencia de comandos dsilog recomendada

En la secuencia de comandos siguiente, un proceso `dsilog` recibe un flujo continuo de datos. `feed_one_data_row` se escribe como una función, que proporciona un flujo de datos continuo a un único proceso `dsilog`.

```
# Begin data feed function
feed_one_data_row()
{
    while :
    do
        # Perform whatever operations necessary to produce one row
        # of data for feed to a dsilog process
        sleep 50
    done
}
# End data feed function

# Script mainline code
feed_one_data_row | dsilog sdlname classname
```

Registro de datos vmstat

En este ejemplo se muestra cómo configurar la integración de orígenes de datos utilizando la configuración predeterminada para registrar los dos primeros valores indicados por `vmstat`. Se puede consultar esta sección como una introducción general del funcionamiento del proceso de integración de orígenes de datos, o realizar cada tarea para crear un archivo de registro DSI equivalente en el sistema.

Los procedimientos necesarios para implementar la integración de orígenes de datos son los siguientes:

- Creación de un archivo de especificación de clase.
- Compilación de un archivo de especificación de clase.
- Inicio del proceso de registro `dsilog`

Creación de un archivo de especificación de clase.

El archivo de especificación de clase es un archivo de texto que se crea para describir la clase, o un conjunto de datos entrantes, así como cada número individual que se pretende registrar como métrica dentro de la clase. El archivo se puede crear con el editor de texto que se prefiera. Debería crearse el archivo para este ejemplo de integración de orígenes de datos en el directorio `/tmp/`.

En el ejemplo siguiente se muestra el archivo de especificación de clase requerido para describir los dos primeros números `vmstat` para su registro en una clase denominada `VMSTAT_STATS`. Como sólo se definen dos métricas en esta clase, el proceso de registro ignora el resto de cada uno de los registros de salida de `vmstat`. Cada línea del archivo se explica en las líneas de comentarios que van después.

```
CLASS VMSTAT_STATS = 10001;
    # Assigns a unique name and number to vmstat class data.
    # The semicolon is required to terminate the class section
    # of the file.

METRICS
    # Indicates that everything that follows is a description
    # of a number (metric) to be logged.

RUN_Q_PROCS = 106;
    # Assigns a unique name and number to a single metric.
    # The semicolon is required to terminate each metric.

BLOCKED_PROCS = 107;
    # Assigns a unique name and number to another metric.
    # The semicolon is required to terminate each metric.
```

Compilación de un archivo de especificación de clase.

Al compilar el archivo de especificación de clase con `sdlcomp`, se comprueba el archivo en busca de errores sintácticos. Si no se encuentra ninguno, `sdlcomp` crea o actualiza un conjunto de archivos de registro para retener los datos de la clase.

Hay que utilizar el nombre de archivo que se le dio al archivo de especificación de clase y, a continuación, especificar un nombre para *nombre_conjunto_de_archivo_de_registro* que permite recordar con facilidad la clase de datos que contiene el archivo de registro. En el ejemplo siguiente de una salida de compilador y de comandos, se utiliza `/tmp/vmstat.spec` como nombre de archivo y `/tmp/VMSTAT_DATA` para el conjunto de archivos de registro.

```
-> sdlcomp /tmp/vmstat.spec /tmp/VMSTAT_DATA

sdlcomp X.01.04
Check class specification syntax.

CLASS VMSTAT_STATS = 10001;

METRICS
RUN_Q_PROCS      = 106;
BLOCKED_PROCS    = 107;

NOTE: Time stamp inserted as first metric by default.

Syntax check successful.

Update SDL VMSTAT_DATA.
Shared memory ID used by vmstat_data=219

Class VMSTAT_STATS successfully added to log file set.
```

En este ejemplo se crea un conjunto de archivos de registro denominado `VMSTAT_DATA` en el directorio `/tmp/`, que incluye un archivo raíz y un archivo de descripción además del archivo de datos. El conjunto de archivos de registro está preparado para aceptar datos registrados. Si hay errores de sintaxis en el archivo de especificación de clase, se muestran mensajes que indican los problemas y no se crea el conjunto de archivos de registro.

Inicio del proceso de registro dsilog

En este momento, ya se puede derivar la salida de `vmstat` directamente al proceso de registro `dsilog`. Se utiliza el comando siguiente:

```
vmstat 60 | dsilog /tmp/VMSTAT_DATA VMSTAT_STATS &
```

Este comando ejecuta `vmstat` cada 60 segundos y envía la salida directamente a la clase `VMSTAT_STATS` en el conjunto de archivos de registro `VMSTAT_DATA`. El comando se ejecuta en segundo plano. También se puede usar `remsh` para alimentar a `vmstat` desde un sistema remoto.

Obsérvese que se genera el mensaje siguiente al inicio del proceso de registro:

```
Metric null has invalid data
Ignore to end of line, metric value exceeds maximum
```

Este mensaje es el resultado de la línea de encabezado de la salida `vmstat` que no puede registrar `dsilog`. Aunque el mensaje aparece en pantalla, `dsilog` sigue ejecutándose y comienza a registrar datos a partir de la primera línea de entrada válida.

Acceso a los datos

El programa `sdlutil` se utiliza para informar del contenido de la clase:

```
sdlutil /tmp/VMSTAT_DATA -stats VMSTAT_STATS
```



De manera predeterminada, los datos se resumirán y se registrarán una vez cada cinco minutos.

Se pueden usar los argumentos de la línea de comandos del programa `extract` para exportar datos de la clase. Por ejemplo:

```
extract -xp -l /tmp/VMSTAT_DATA -C VMSTAT_STATS -ut -f stdout
```

Hay que tener en cuenta que para exportar datos DSI, hay que ser el usuario raíz o el creador del archivo de registro.

Registro de datos sar desde un archivo

En este ejemplo se muestra cómo configurar varias recopilaciones de datos DSI con la utilidad `sar` (informe de actividad del sistema) estándar para proporcionar los datos.

Cuando se utiliza una utilidad del sistema, es importante comprender exactamente cómo informa los datos esta utilidad. Por ejemplo, obsérvese la diferencia entre los dos comandos `sar` siguientes:

```
sar -u 1 1
```

```
HP-UX hpptc99 A.11.00 E 9000/855 04/10/99
```

```
10:53:15 %usr %sys %wio %idle
10:53:16 2 7 6 85
```

```
sar -u 5 2
```

```
HP-UX hpptc99 A.11.00 E 9000/855 04/10/99
```

```
10:53:31 %usr %sys %wio %idle
10:53:36 4 5 0 91
10:53:41 0 0 0 99
```

```
Average 2 2 0 95
```

Como se puede ver, si se especifica un valor de iteración mayor que 1, `sar` mostrará un promedio en el intervalo. Este promedio puede ser o no de interés pero puede afectar al archivo de especificación de clases DSI y la conversión de datos. Hay que tener en cuenta que la salida de `sar`, o de otras utilidades del sistema, puede ser diferente cuando se ejecuta en distintas plataformas UNIX. Es aconsejable familiarizarse con la utilidad que se está pensando usar antes de crear el archivo de especificación de clase DSI.

Nuestro primer ejemplo usa `sar` para monitorizar la utilización de CPU a través de la opción `-u` de `sar`. Si se consulta la página del manual referida a `sar`, se verá que la opción `-u` informa sobre la parte de tiempo que se ejecuta en modo usuario (`%usr`), ejecutándose en modo de sistema (`%sys`), inactivo con algunos procesos en espera para bloquear la E/S (`%wio`) y, por lo demás, inactivo (`%idle`). Como se pretende monitorizar la actividad de la CPU durante un período prolongado de tiempo, se utiliza la forma de `sar` que no muestre el promedio.

Creación de un archivo de especificación de clase

La primera tarea que se va a completar es la creación de un archivo de especificación de clase DSI. A continuación se muestra un ejemplo de una especificación de clase que se puede usar para describir los datos entrantes:

```
# sar_u.spec
#
# sar -u class definition for HP systems.
#
# ==> 1 minute data; max 24 hours; indexed by hour; roll by day
#

CLASS sar_u = 1000
LABEL "sar -u data"
INDEX BY      hour
MAX INDEXES   24
ROLL BY       day
ACTION "./sar_u_roll $PT_START$ $PT_END$"
RECORDS PER HOUR 60
;

METRICS

hours_1 = 1001
LABEL "Collection Hour"
PRECISION 0;

minutes_1 = 1002
LABEL "Collection Minute"
PRECISION 0;

seconds_1 = 1003
LABEL "Collection Second"
PRECISION 0;

user_cpu = 1004
LABEL "%user"
AVERAGED
MAXIMUM 100
PRECISION 0
;

sys_cpu = 1005
LABEL "%sys"
AVERAGED
MAXIMUM 100
PRECISION 0
```



```

;

wait_IO_cpu = 1006
LABEL "%wio"
AVERAGED
MAXIMUM 100
PRECISION 0
;

idle_cpu = 1007
LABEL "%idle"
AVERAGED
MAXIMUM 100
PRECISION 0
;

```

Compilación de un archivo de especificación de clase

La tarea siguiente consiste en compilar el archivo de especificación de clase usando el comando siguiente.

```
sdlcomp sar_u.spec sar_u_log
```

La salida del comando `sar -u` es una línea de encabezado del sistema, una línea en blanco, una línea de encabezado de opción y una línea de datos que consta de una marca de hora seguida por los datos que se desean capturar. La última línea es la única que interesa. Por tanto, en el comando `sar -u`, se precisa de un mecanismo para guardar únicamente la última línea de salida y alimentar esos datos en DSI.

`dsilog` espera recibir datos de `stdin`. Para comenzar el proceso de registro, hay que canalizar la salida del proceso que se está usando a `dsilog`. Sin embargo, sólo se puede tener una barra vertical (`|`) en la línea de comandos. Cuando se usan dos barras verticales, el proceso de búfer de UNIX conserva la salida del primer comando hasta que se hayan escrito 8.000 caracteres antes de continuar con el segundo comando y canalizar la salida al archivo de registro. Como resultado, escribir algo como lo siguiente no sirve:

```
sar -u 60 1 | tail -1 | dsilog
```

Por consiguiente, hay que usar un `fifo` como origen de entrada para DSI. Sin embargo, esto no se realiza sin problemas.

Por ejemplo, se va a usar la siguiente secuencia de comandos:

```
#!/bin/ksh                                sar_u_feed

# sar_u_feed script that provides sar -u data to DSI via
# a fifo(sar_u.fifo)

while :                                     # (infinite loop)
do

# specify a one minute interval using tail to extract the
# last sar output record(contains the time stamp and data),
# saving the data to a file.

/usr/bin/sar -u 60 1 2>/tmp/dsierr | tail -1 > /usr/tmp/sar_u_data

```

```

# Copy the sar data to the fifo that the dsilog process is
# reading.

cat /usr/tmp/sar_u_data > ./sar_u.fifo

done

```

Desafortunadamente, esta secuencia de comandos no producirá los resultados deseados si se ejecuta tal cual. Esto es debido a que el comando `cat` abre el `fifo`, escribe el registro de datos y, a continuación, cierra el `fifo`. El cierre indica a `dsilog` que no hay más datos que escribir en el registro, por tanto `dsilog` escribe este registro de datos y finaliza. Es necesario por tanto un proceso ficticio para que "retenga" abierto el `fifo`. Por consiguiente, se necesita un `fifo` ficticio y un proceso que abra el `fifo` ficticio para la entrada y el `sar_u.fifo` para la salida. De esta forma se retendrá abierto el `sar_u.fifo`, evitando que finalice `dsilog`.

Inicio del proceso de registro DSI

A continuación se describe un método paso a paso para obtener los datos de `sar -u` en `dsilog`.

- 1 Cree dos `fifo`; uno será el `fifo` ficticio utilizado para "mantener abierto" el `fifo` de entrada real.

```

# Dummy fifo.
mkfifo ./hold_open.fifo
# Real input fifo for dsilog
mkfifo ./sar_u.fifo

```

- 2 Inicie `dsilog` usando la opción `-i` para especificar la entrada que procede de un `fifo`. Es importante iniciar `dsilog` antes de iniciar la fuente de datos `sar (sar_u_feed)`.

```

dsilog ./sar_u_log sar_u \
-i ./sar_u.fifo &

```

- 3 Inicie el proceso ficticio para mantener abierto el `fifo` de entrada.

```

cat ./hold_open.fifo \
> ./sar_u.fifo &

```

- 4 Inicie la secuencia de comandos de la fuente de datos `sar (sar_u_feed)`.

```

./sar_u_feed &

```

- 5 La secuencia de comandos `sar_u_feed` pasará datos a `dsilog` hasta que se cierre o hasta que se cierre el `cat` que retiene abierto el `fifo`. El archivo de especificación de clase indica que `sar_u_log` se indexará por hora, contiene un máximo de 24 horas y que al comienzo del día siguiente (roll by day) se ejecutará la secuencia de comandos `sar_u_roll`.

```

#!/bin/ksh                sar_u_roll
#
# Save parameters and current date in sar_u_log_roll_file.
# (Example of adding comments/other data to the roll file).

mydate=`date`
echo "$# $0 $1 $2" >> ./sar_u_log_roll_file
echo $mydate          >> ./sar_u_log_roll_file

```

```
extract -l ./sar_u_log -C sar_u -B $1 -E $2 -1 -f \  
stdout -xp >> ./sar_u_log_roll_file
```

- 6 La secuencia de comandos de supresión guarda los datos que se están suprimiendo en un archivo de texto ASCII, que se puede revisar con un editor de texto o imprimirse.

Registro de datos sar desde varios archivos

Si se está interesado en más de una única utilización de CPU, se puede tener un archivo de especificación de clase que describa los datos o tener un archivo de especificación de clase para cada opción y compilarlos en un conjunto de archivos de registro. En el primer ejemplo se muestran archivos de especificación de clase independientes compilados en un único conjunto de archivos de registro.

En este ejemplo, se va a monitorizar la utilización de la CPU, la actividad del búfer (`sar -b`) y las llamadas del sistema (`sar -c`). El registro de los datos de esta forma requiere tres archivos de especificación de clase, tres procesos `dsilog`, tres `fifo` de entrada de `dsilog` y tres secuencias de comandos para proporcionar los datos de `sar`.

Creación de archivos de especificación de clase

A continuación se muestran los archivos de especificación de clase para cada una de estas opciones.

```
# sar_u_mc.spec  
#  
# sar -u class definition for log files on HP systems.  
#  
# ==> 1 minute data; max 24 hours; indexed by hour; roll by day  
#  
CLASS sar_u = 1000  
LABEL "sar -u data"  
INDEX BY          hour  
MAX INDEXES      24  
ROLL BY          day  
ACTION "./sar_u_mc_roll $PT_START$ $PT_END$"  
RECORDS PER HOUR 60  
;  
  
METRICS  
  
hours_1 = 1001  
LABEL "Collection Hour"  
PRECISION 0  
;  
  
minutes_1 = 1002  
LABEL "Collection Minute"  
PRECISION 0  
;
```

```

seconds_1 = 1003
LABEL "Collection Second"
PRECISION 0
;
user_cpu = 1004
LABEL "%user"
AVERAGED
MAXIMUM 100
PRECISION 0
;

sys_cpu = 1005
LABEL "%sys"
AVERAGED
MAXIMUM 100
PRECISION 0
;

wait_IO_cpu = 1006
LABEL "%wio"
AVERAGED
MAXIMUM 100
PRECISION 0
;

idle_cpu = 1007
LABEL "%idle"
AVERAGED
MAXIMUM 100
PRECISION 0
;

# sar_b_mc.spec
#
# sar -b class definition for log files on HP systems.
#
# ==> 1 minute data; max 24 hours; indexed by hour; roll by day
#

CLASS sar_b = 2000
LABEL "sar -b data"
INDEX BY          hour
MAX INDEXES       24
ROLL BY           day
ACTION "./sar_b_mc_roll $PT_START$ $PT_END$"
RECORDS PER HOUR 60
;

METRICS

hours_2 = 2001
LABEL "Collection Hour"
PRECISION 0
;

```

```

minutes_2 = 2002
LABEL "Collection Minute"
PRECISION 0
;

seconds_2 = 2003
LABEL "Collection Second"
PRECISION 0
;

bread_per_sec = 2004

LABEL "bread/s"
PRECISION 0
;

lread_per_sec = 2005
LABEL "lread/s"
PRECISION 0
;

read_cache = 2006
LABEL "%rcache"
MAXIMUM 100
PRECISION 0
;

bwrit_per_sec = 2007
LABEL "bwrit/s"
PRECISION 0
;

lwrit_per_sec = 2008
LABEL "lwrit/s"
PRECISION 0
;

write_cache = 2009
LABEL "%wcache"
MAXIMUM 100
PRECISION 0
;

pread_per_sec = 2010
LABEL "pread/s"
PRECISION 0
;

pwrit_per_sec = 2011
LABEL "pwrit/s"
PRECISION 0
;

# sar_c_mc.spec
#

```

```

# sar -c class definition for log files on HP systems.
#
# ==> 1 minute data; max 24 hours; indexed by hour; roll by day
#

CLASS sar_c = 5000
LABEL "sar -c data"
INDEX BY      hour
MAX INDEXES   24
ROLL BY       day
ACTION "./sar_c_mc_roll $PT_START$ $PT_END$"
RECORDS PER HOUR 60

;

METRICS

hours_5 = 5001
LABEL "Collection Hour"
PRECISION 0
;

minutes_5 = 5002
LABEL "Collection Minute"
PRECISION 0
;

seconds_5 = 5003
LABEL "Collection Second"
PRECISION 0
;

scall_per_sec = 5004
LABEL "scall/s"
PRECISION 0
;

sread_per_sec = 5005
LABEL "sread/s"
PRECISION 0
;

swrit_per_sec = 5006
LABEL "swrit/s"
PRECISION 0
;

fork_per_sec = 5007
LABEL "fork/s"
PRECISION 2
;

exec_per_sec = 5008
LABEL "exec/s"
PRECISION 2

```

```

;

rchar_per_sec = 5009
LABEL "rchar"
PRECISION 0
;

wchar_per_sec = 5010
LABEL "wchar/s"
PRECISION 0
;

```

The following are the two additional scripts that are needed to supply the sar data.

```

#!/bin/ksh

# sar_b_feed script that provides sar -b data to DSI via
# a fifo (sar_b.fifo)

while :                               # (infinite loop)
do

# specify a one minute interval using tail to extract the
# last sar output record(contains the time stamp and data),
# saving the data to a file.

/usr/bin/sar -b 60 1 2>/tmp/dsierr | tail -1 &> \
/usr/tmp/sar_b_data

# Copy the sar data to the fifo that the dsilog process is reading.
cat /usr/tmp/sar_b_data > ./sar_b.fifo

done

#!/bin/ksh                               sar_c_feed

# sar_c_feed script that provides sar -c data to DSI via
# a fifo(sar_c.fifo)

while :                               # (infinite loop)
do

# specify a one minute interval using tail to extract the
# last sar output record(contains the time stamp and data),
# saving the data to a file.

/usr/bin/sar -c 60 1 2>/tmp/dsierr | tail -1 > /usr/tmp/sar_c_data

# Copy the sar data to the fifo that the dsilog process is reading.

```

```
cat /usr/tmp/sar_c_data > ./sar_c.fifo  
  
done
```

Compilación de archivos de especificación de clase

Se compilan los tres archivos de especificación en un conjunto de archivos de registro:

```
sdlcomp ./sar_u_mc.spec sar_mc_log  
sdlcomp ./sar_b_mc.spec sar_mc_log  
sdlcomp ./sar_c_mc.spec sar_mc_log
```

Inicio del proceso de registro DSI

Volviendo al método paso a paso de los datos sar:

- 1 Cree cuatro fifo; uno será el fifo ficticio utilizado para "mantener abiertos" los tres fifo de entrada reales.

```
# Dummy fifo.  
mkfifo ./hold_open.fifo  
  
# sar -u input fifo for dsilog.  
mkfifo ./sar_u.fifo  
  
# sar -b input fifo for dsilog.  
mkfifo ./sar_b.fifo  
  
# sar -c input fifo for dsilog.  
mkfifo ./sar_c.fifo
```

- 2 Inicie dsilog usando la opción `-i` para especificar la entrada que procede de un fifo. Es importante iniciar dsilog antes de iniciar las fuentes de datos sar.

```
dsilog ./sar_mc_log sar_u \  
-i ./sar_u.fifo &  
  
dsilog ./sar_mc_log sar_b \  
-i ./sar_b.fifo &  
  
dsilog ./sar_mc_log sar_c \  
-i ./sar_c.fifo &
```

- 3 Inicie el proceso ficticio para mantener abierto el fifo de entrada.

```
cat ./hold_open.fifo \  
> ./sar_u.fifo &  
  
cat ./hold_open.fifo \  
> ./sar_b.fifo &  
  
cat ./hold_open.fifo \  
> ./sar_c.fifo &
```


4 Inicie las secuencias de comandos de las fuentes de datos sar.

```
./sar_u_feed &
```

```
./sar_b_feed &
```

```
./sar_c_feed &
```

Registro de datos sar para varias opciones

En el último ejemplo de utilización de sar para proporcionar datos a DSI se utiliza un archivo de especificación para definir los datos desde varias opciones sar (ubycwvm).

```
# sar_ubycwvm.spec
#
# sar -ubycwvm class definition for HP systems.
#
# ==> 1 minute data; max 24 hours; indexed by hour; roll by day
#

CLASS sar_ubycwvm = 1000
LABEL "sar -ubycwvm data"
INDEX BY          hour
MAX INDEXES       24
ROLL BY           day
ACTION "./sar_ubycwvm_roll $PT_START$ $PT_END$"
RECORDS PER HOUR 60
;

METRICS
hours = 1001
LABEL "Collection Hour"
PRECISION 0;

minutes = 1002
LABEL "Collection Minute"
PRECISION 0;

seconds = 1003
LABEL "Collection Second"
PRECISION 0;

user_cpu = 1004
LABEL "%user"
AVERAGED
MAXIMUM 100
PRECISION 0
;

sys_cpu = 1005
LABEL "%sys"
AVERAGED
MAXIMUM 100
```

```
PRECISION 0
;

wait_IO_cpu = 1006
LABEL "%wio"
AVERAGED
MAXIMUM 100
PRECISION 0
;

idle_cpu = 1007
LABEL "%idle"
AVERAGED
MAXIMUM 100
PRECISION 0
;

bread_per_sec = 1008
LABEL "bread/s"
PRECISION 0
;

lread_per_sec = 1009
LABEL "lread/s"
PRECISION 0
;

read_cache = 1010
LABEL "%rcache"
MAXIMUM 100
PRECISION 0
;

bwrit_per_sec = 1011
LABEL "bwrit/s"
PRECISION 0
;

lwrit_per_sec = 1012
LABEL "lwrit/s"
PRECISION 0
;

write_cache = 1013
LABEL "%wcache"
MAXIMUM 100
PRECISION 0
;

pread_per_sec = 1014
LABEL "pread/s"
PRECISION 0
;

pwrit_per_sec = 1015
LABEL "pwrit/s"
```

```
PRECISION 0
;

rawch = 1016
LABEL "rawch/s"
PRECISION 0
;

canch = 1017
LABEL "canch/s"

PRECISION 0
;

outch = 1018
LABEL "outch/s"
PRECISION 0
;

rcvin = 1019
LABEL "rcvin/s"
PRECISION 0
;

xmtin = 1020
LABEL "xmtin/s"
PRECISION 0
;

mdmin = 1021
LABEL "mdmin/s"
PRECISION 0
;

scall_per_sec = 1022
LABEL "scall/s"
PRECISION 0
;

sread_per_sec = 1023
LABEL "sread/s"
PRECISION 0
;

swrit_per_sec = 1024
LABEL "swrit/s"
PRECISION 0
;

fork_per_sec = 1025
LABEL "fork/s"
PRECISION 2
;

exec_per_sec = 1026
```

```
LABEL "exec/s"
PRECISION 2
;

rchar_per_sec = 1027
LABEL "rchar/s"
PRECISION 0
;

wchar_per_sec = 1028
LABEL "wchar/s"

PRECISION 0
;

swpin = 1029
LABEL "swpin/s"
PRECISION 2
;

bswin = 1030
LABEL "bswin/s"
PRECISION 1
;

swpot = 1031
LABEL "swpot/s"
PRECISION 2
;

bswot = 1032
LABEL "bswot/s"
PRECISION 1
;

blks = 1033
LABEL "pswch/s"
PRECISION 0
;

iget_per_sec = 1034
LABEL "iget/s"
PRECISION 0
;

namei_per_sec = 1035
LABEL "namei/s"
PRECISION 0
;

dirbk_per_sec = 1036
LABEL "dirbk/s"
PRECISION 0
;

num_proc = 1037
```

```

LABEL "num proc"
PRECISION 0
;

proc_tbl_size = 1038
LABEL "proc tbl size"
PRECISION 0
;

proc_ov = 1039
LABEL "proc ov"
PRECISION 0

;

num_inode = 1040
LABEL "num inode"
PRECISION 0
;

inode_tbl_sz = 1041
LABEL "inode tbl sz"
PRECISION 0
;

inode_ov = 1042
LABEL "inode ov"
PRECISION 0
;

num_file = 1043
LABEL "num file"
PRECISION 0
;

file_tbl_sz = 1044
LABEL "file tbl sz"
PRECISION 0
;

file_ov = 1045
LABEL "file ov"
PRECISION 0
;

msg_per_sec = 1046
LABEL "msg/s"
PRECISION 2
;

LABEL "sema/s"
PRECISION 2
;

```

En este punto, es preciso revisar la salida generada de

```
sar -ubycwvm 1 1:
HP-UX hpptc16 A.09.00 E 9000/855    04/11/95

12:01:41   %usr   %sys   %wio   %idle
          bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s
          pwrit/s
          rawch/s canch/s outch/s   cvin/s  xmtin/s  madmin/s
          scall/s sread/s swrit/s   fork/s  exec/s   rchar/s wchar/s
          swpin/s bswin/s swpot/s  bswot/s  pswch/s
          iget/s namei/s dirbk/s
          text-sz  ov  proc-sz  ov  inod-sz  ov  file-sz  ov
          msg/s   sema/s

12:01:42    22     48     30     0
           0    342    100    33     81     59     0     0
           0     0    470     0     0     0
           801    127     71     1.00  1.00  975872 272384
           0.00  0.0    0.00  0.0    251
           28    215    107
          N/A  N/A  131/532    0  639/644    0  358/1141    0
          40.00  0.00
```

Esta salida es similar a la salida de `sar -u` con varias líneas adicionales de encabezados y datos. Se usará de nuevo `tail` para extraer las líneas de datos, pero es preciso presentarlo como "un único" registro de datos a `dsilog`. La secuencia de comandos siguiente captura los datos y usa la utilidad `tr` (traducir carácter) para "quitar" las fuentes de líneas para que `dsilog` las considere como una única línea de datos de entrada.

```
#!/bin/ksh                               Sar_ubycwvm_feed

# Script that provides sar data to DSI via a fifo(sar_data.fifo)

while :                                   # (infinite loop)
do

# specify a one minute interval using tail to extract the
# last sar output records (contains the time stamp and data)
# and pipe that data to tr to strip the new lines converting
# the eight lines of output to one line of output.

/usr/bin/sar -ubycwvm 60 1 2>/tmp/dsierr | tail -8 | \
tr "\012" " " > /usr/tmp/sar_data

# Copy the sar data to the fifo that the dsilog process is reading.

cat /usr/tmp/sar_data > ./sar_data.fifo

# Print a newline on the fifo so that DSI knows that this is
# the end of the input record.

print "\012" > ./sar_data.fifo

done
```

El proceso paso a paso sigue el del ejemplo `sar -u`, con la excepción de los nombres de conjuntos de archivos de registro, nombres de clases, nombre de `fifo` (`sar_ubycwvm.fifo`) y la secuencia de comandos listada anteriormente para proporcionar datos `sar`.

Registro del número de usuarios del sistema

En el ejemplo siguiente se utiliza `who` para monitorizar el número de usuarios del sistema. De nuevo, se comienza con un archivo de especificación de clase.

```
# who_wc.spec
#
# who word count DSI spec file
#

CLASS who_metrics = 150
LABEL "who wc data"
INDEX BY          hour
MAX INDEXES      120
ROLL BY          hour
RECORDS PER HOUR 60
;

METRICS
who_wc = 151
label "who wc"
averaged
maximum 1000
precision 0
;
```

Se compila el archivo de especificación para crear un archivo de registro:

```
sdlcomp ./who_wc.spec ./who_wc_log.
```

A diferencia de `sar`, no se puede especificar un valor de intervalo o iteración con `who`, por tanto se crea una secuencia de comandos que proporciona, como mínimo, control de intervalos.

```
#!/bin/ksh          who_data_feed

while :
do
    # sleep for one minute (this should correspond with the
    # RECORDS PER HOUR clause in the specification file).

    sleep 60

    # Pipe the output of who into wc to count
    # the number of users on the system.

    who | wc -l > /usr/tmp/who_data

    # copy the data record to the pipe being read by dsilog.
```

```
cat /usr/tmp/who_data > ./who.fifo
```

```
done
```

De nuevo, se necesita un `fifo` y una secuencia de comandos para suministrar datos a `dsilog`, por tanto se vuelve al proceso paso a paso.

- 1 Cree dos `fifo`; uno será el `fifo` ficticio utilizado para "mantener abierto" el `fifo` de entrada real.

```
# Dummy fifo.  
mkfifo ./hold_open.fifo
```

```
# Real input fifo for dsilog.  
mkfifo ./who.fifo
```

- 2 Inicie `dsilog` usando la opción `-i` para especificar la entrada que procede de un `fifo`. Es importante iniciar `dsilog` antes de iniciar las fuentes de datos `who`.

```
dsilog ./who_wc_log who_metrics \  
-i ./who.fifo &
```

- 3 Inicie el proceso ficticio para mantener abierto el `fifo` de entrada.

```
cat ./hold_open.fifo \  
> ./who.fifo &
```

- 4 Inicie la secuencia de comandos de la fuente de datos `who` (`who_data_feed`).

```
./who_data_feed &
```


17 Mensajes de error

Hay tres tipos de mensajes de error de DSI: especificación de clase, proceso de registro `dsilog` y errores generales.

- El formato de los mensajes de error de especificación de clase consiste en el prefijo `SDL` seguido del número de mensaje.
- El formato de los mensajes de proceso de registro `dsilog` consiste en el prefijo `DSILOG` seguido del número de mensaje.
- Los mensajes de error generales se pueden generar como consecuencia de cualquiera de los anteriores así como por otras tareas. Estos mensajes llevan un prefijo de signo menos (-) y el número de mensaje.

Los mensajes de error de DSI se enumeran en este capítulo. Los mensajes de error `SDL` y `DSILOG` se listan en orden numérico, junto con las acciones que hay que tomar para resolver la condición de error. Los mensajes de error generales son autoexplicativos, por tanto no se proporcionan acciones de recuperación.

Mensajes de error SDL

Los mensajes de error `SDL` son mensajes de error de la especificación de clase Self Describing Logfile (Archivo de registro autodescriptivo), con el formato `SDL<número_de_mensajes>`.

Mensaje SDL1

ERROR: Expected equal sign, "=".

Se esperaba aquí un "=".

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL2

ERROR: Expected semi-colon, ";".

Un punto y coma (;) marca el final de la especificación de clase y el final de cada especificación de métrica. También puede ver este mensaje si se encuentran palabras incorrectas o con errores ortográficos donde debería haber un punto y coma.

Por ejemplo: si se introduce

```
class xxxxx = 10
  label "this is a test"
  metric 1000;
```

en lugar de

```
class xxxxx = 10
  label "this is a test"
  capacity 1000;
```

se mostrará este mensaje de error apuntando a la palabra "metric".

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL3

ERROR: Precision must be one of {0, 1, 2, 3, 4, 5}

La precisión determina el número de lugares decimales utilizados al pasar internamente números a enteros y volverlos a representaciones numéricas del valor de la métrica.

Acción: Consulte [PRECISION](#) en Capítulo 14 para obtener más información.

Mensaje SDL4

ERROR: Expected quoted string.

Se esperaba una cadena de texto.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL5

ERROR: Unterminated string.

La cadena debe terminar con dobles comillas.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL6

NOTE: Time stamp inserted at first metric by default.

Una métrica de marca de hora se inserta automáticamente como la primera métrica de cada clase.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL7

ERROR: Expected metric description.

La sección de métrica deberá comenzar con la palabra clave METRICS antes de la primera definición de métrica.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL8

ERROR: Expected data class specification.

La sección de clase de la especificación de clase debe comenzar con la palabra clave CLASS.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL9

ERROR: Expected identifier.

Se esperaba un identificador para la métrica o la clase. El identificador debe comenzar con un carácter alfabético, puede contener caracteres alfanuméricos o subrayados y no distingue entre mayúsculas y minúsculas.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL10

ERROR: Expected positive integer.

El formato numérico no es correcto.

Acción: Hay que introducir únicamente enteros positivos.

Mensaje SDL13

ERROR: Expected specification for maximum number of indexes.

Se requiere el número máximo de índices para calcular la capacidad de la clase.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL14

ERROR: Syntax Error.

La sintaxis introducida no es correcta.

Acción: Comprobar la sintaxis y realizar las correcciones necesarias. Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL15

ERROR: Expected metric description.

Falta una descripción de la métrica.

Acción: Hay que introducir una descripción de métrica para definir los elementos de datos individuales para la clase. Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL16

ERROR: Expected metric type.

Cada métrica debe tener un *nombre_de_métrica* y un *id_de_métrica* numérico.

Acción: Consulte [Descripciones de métrica](#) en Capítulo 14 para obtener más información.

Mensaje SDL17

ERROR: Time stamp metric attributes may not be changed.

Una métrica timestamp se inserta automáticamente como la primera métrica de cada clase. Se puede cambiar la posición de la marca de hora o eliminarla y usar una marca de hora UNIX.

Acción: Consulte [Descripciones de métrica](#) en Capítulo 14 para obtener más información.

Mensaje SDL18

ERROR: Roll action limited to 199 characters.

El límite superior para la acción ROLL BY es de 199 caracteres.

Acción: Consulte [Parámetros INDEX BY, MAX INDEXES y ROLL BY](#) en Capítulo 14 para obtener más información.

Mensaje SDL19

ERROR: Could not open specification file (file).

En la línea de comandos `sdlcomp specification_file`, no se puede abrir el archivo de especificación. El error sigue en la línea siguiente de la siguiente forma:

```
$/usr/perf/bin/sdlcomp /xxx
```

ERROR: Could not open specification file /xxx.

Acción: Comprobar que el archivo se puede leer. Si es así, se comprueba el nombre del archivo y que se ha introducido correctamente.

Mensaje SDL20

ERROR: Metric descriptions not found.

La descripción de la métrica tiene un formato incorrecto.

Acción: Hay que asegurarse que se comienza la sección de métricas de la instrucción de la clase con la palabra clave METRICS. Consulte [Descripciones de métrica](#) en Capítulo 14 para obtener más información.

Mensaje SDL21

ERROR: Expected metric name to begin metric description.

Puede faltar el nombre de la métrica o la descripción de la métrica está incorrectamente formateada.

Acción: Puede faltar el nombre de la métrica o la descripción de la métrica está incorrectamente formateada.

Mensaje SDL24

ERROR: Expected MAX INDEXES specification.

Se requiere un valor MAX INDEXES cuando se especifica INDEX BY.

Acción: Hay que introducir el valor requerido. Consulte [Parámetros INDEX BY, MAX INDEXES y ROLL BY](#) en Capítulo 14 para obtener más información.

Mensaje SDL25

ERROR: Expected index SPAN specification.

Falta un valor para INDEX BY.

Acción: Hay que introducir un calificador al especificar INDEX BY. Consulte [Parámetros INDEX BY, MAX INDEXES y ROLL BY](#) en Capítulo 14 para obtener más información.

Mensaje SDL26

ERROR: Minimum must be zero.

El número debe ser un cero o un valor superior.

Mensaje SDL27

Expected positive integer.

Falta un valor positivo.

Acción: Hay que introducir únicamente enteros positivos.

Mensaje SDL29

ERROR: Summarization metric does not exist.

Se ha usado SUMMARIZED BY como método de resumen, pero no se especificó un *nombre_de_métrica*.

Acción: Consulte [Descripciones de métrica](#) en Capítulo 14 para obtener más información.

Mensaje SDL30

ERROR: Expected 'HOURL', 'DAY', or 'MONTH'.

Falta un calificador en la entrada.

Acción: Hay que introducir uno de estos calificadores. Consulte [Parámetros INDEX BY, MAX INDEXES y ROLL BY](#) en Capítulo 14 para obtener más información.

Mensaje SDL33

ERROR: Class id number must be between 1 and 999999.

El class-id debe ser un valor numérico y puede contener hasta 6 dígitos.

Acción: Hay que introducir un número de ID de clase para la clase y no debe exceder el máximo de seis dígitos. Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL35

ERROR: Found more than one index/capacity statement.

Sólo se puede tener una instrucción INDEX BY o CAPACITY por sección CLASS.

Acción: Hay que completar las entradas de acuerdo con las restricciones de formato de [Sintaxis de especificación de clase](#) en Capítulo 14.

Mensaje SDL36

ERROR: Found more than one metric type statement.

Sólo se puede tener una palabra clave METRICS para cada definición de métrica.

Acción: Consulte [Descripciones de métrica](#) en Capítulo 14 para obtener información sobre el formato.

Mensaje SDL37

ERROR: Found more than one metric maximum statement.

Sólo se puede tener una instrucción MAXIMUM para cada definición de métrica.

Acción: Consulte [Descripciones de métrica](#) en Capítulo 14 para obtener información sobre el formato.

Mensaje SDL39

ERROR: Found more than one metric summarization specification.

Sólo se puede tener un método de resumen (TOTALLED, AVERAGED o SUMMARIZED BY) para cada definición de métrica.

Acción: Consulte [Método de resumen](#) en Capítulo 14 para obtener más información.

Mensaje SDL40

ERROR: Found more than one label statement.

Sólo se puede tener una LABEL para cada definición de métrica o de clase.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL42

ERROR: Found more than one metric precision statement.

Sólo se puede tener una instrucción PRECISION para cada definición de métrica.

Acción: Consulte [PRECISION](#) en Capítulo 14 para obtener más información.

Mensaje SDL44

ERROR: SCALE, MINIMUM, MAXIMUM, (summarization) are inconsistent with text metrics

Estos elementos de la sintaxis de la especificación de clase sólo son válidos para las métricas numéricas.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL46

ERROR: Inappropriate summarization metric (!).

No se puede resumir por la métrica de marca de hora.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL47

ERROR: Expected metric name.

Cada instrucción METRICS debe incluir un *nombre_de_métrica*.

Acción: Consulte [Descripciones de métrica](#) en Capítulo 14 para obtener más información.

Mensaje SDL47

ERROR: Expected metric name.

Cada instrucción METRICS debe incluir un *nombre_de_métrica*.

Acción: Consulte [Descripciones de métrica](#) en Capítulo 14 para obtener más información.

Mensaje SDL48

ERROR: Expected positive integer.

La instrucción CAPACITY requiere un entero positivo.

Acción: Consulte [CAPACITY](#) en Capítulo 14 para obtener más información.

Mensaje SDL49

ERROR: Expected metric specification statement.

La palabra clave METRICS debe preceder a la primera definición de métrica.

Acción: Consulte [Descripciones de métrica](#) en Capítulo 14 para obtener más información.

Mensaje SDL50

Object name too long.

El *nombre_de_métrica* o *nombre_de_clase* sólo pueden tener hasta 20 caracteres.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL51

ERROR: Label too long (max 20 chars).

La *etiqueta_de_clase* o *etiqueta_de_métrica* sólo pueden tener hasta 20 caracteres.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL53

ERROR: Metric must be between 1 and 999999.

El *id_de_métrica* sólo puede contener hasta 6 dígitos.

Acción: Consulte [Descripciones de métrica](#) en Capítulo 14 para obtener más información.

Mensaje SDL54

ERROR: Found more than one collection rate statement.

Sólo se puede tener una instrucción RECORDS PER HOUR para cada descripción de clase.

Acción: Consulte [RECORDS PER HOUR](#) en Capítulo 14 para obtener más información.

Mensaje SDL55

ERROR: Found more than one roll action statement.

Sólo se puede tener una instrucción ROLL BY para cada especificación de clase.

Acción: Consulte [Parámetros INDEX BY, MAX INDEXES y ROLL BY](#) en Capítulo 14 para obtener más información.

Mensaje SDL56

ERROR: ROLL BY option cannot be specified without INDEX BY option.

La instrucción ROLL BY debe ir precedida por una instrucción INDEX BY.

Acción: Consulte [Parámetros INDEX BY, MAX INDEXES y ROLL BY](#) en Capítulo 14 para obtener más información.

Mensaje SDL57

ERROR: ROLL BY must specify time equal to or greater than INDEX BY.

Como el intervalo de supresión depende del intervalo de índice para identificar los datos que se van a descartar, el tiempo de ROLL BY debe ser mayor o igual que el tiempo de INDEX BY.

Acción: Consulte [Parámetros INDEX BY, MAX INDEXES y ROLL BY](#) en Capítulo 14 para obtener más información.

Mensaje SDL58

ERROR: Metric cannot be used to summarize itself.

La métrica SUMMARIZED BY no puede ser igual que *nombre_de_métrica*.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL62

ERROR: Could not open SDL (name).

Mensajes descriptivos siguen a este error. Podría ser un error del sistema de archivos como en:

```
$/usr/perf/bin/sdlutil xxxxxx -classes
ERROR: Could not open SDL xxxxxx.
ERROR: Could not open log file set.
```

o podría ser un error interno como en:

```
$/usr/perf/bin/sdlutil xxxxxx -classes
ERROR: Could not open SDL xxxxxx.
ERROR: File is not SDL root file or the
description file is not accessible.
```

Es posible que también se vea este error si el archivo de registro se ha movido. Como la información del nombre de la ruta se almacena en los archivos de registro DSI, los archivos de registro no se pueden mover a otros directorios.

Acción: Si la descripción anterior o los mensajes de seguimiento no apuntan a algún problema obvio, se utiliza sdlutil para suprimir el conjunto de archivos de registro y volver a crearlo.

Mensaje SDL63

ERROR: Some files in log file set (name) are missing.

Se comprobó la lista de archivos que forman el conjunto de archivos de registro y no se encontraron uno o varios archivos necesarios para realizar correctamente la operación.

Acción: A no ser que se sepa precisamente lo que ha ocurrido, la mejor acción es utilizar sdlutil para suprimir el conjunto de archivos de registro y comenzar de nuevo.

Mensaje SDL66

ERROR: Could not open class (name).

A continuación aparece un mensaje descriptivo.

Acción: A no ser que sea evidente el motivo del problema, la mejor acción es utilizar `sdlutil` para suprimir el conjunto de archivos de registro y comenzar de nuevo.

Mensaje SDL67

ERROR: Add class failure.

A continuación aparecen mensajes descriptivos.

El compilador no pudo agregar la nueva clase al conjunto de archivos de registro.

Acción: Si son accesibles todas las clases correctas en el conjunto de archivos de registro, especifique un nuevo conjunto de archivos de registro u otro diferente. Si no lo son, hay que utilizar `sdlutil` para suprimir el conjunto de archivos de registro y comenzar de nuevo.

Mensaje SDL72

ERROR: Could not open export files (name).

No se pudo abrir el archivo al que se supone que se van a escribir los datos exportados.

Acción: Hay que ver si existe la ruta de archivo de exportación y qué permisos tiene.

Mensaje SDL73

ERROR: Could not remove shared memory ID (name).

A continuación aparece un mensaje descriptivo.

Acción: El usuario que ha creado el conjunto de archivos de registro o el usuario raíz es el que puede suprimir el ID de memoria compartida. Hay que utilizar el comando UNIX `ipcrm -m id` para suprimir el ID de memoria compartida.

Mensaje SDL74

ERROR: Not all files could be removed.

No se pueden suprimir todos los archivos del conjunto de archivos de registro.

A continuación aparecen mensajes descriptivos.

Acción: Hay que hacer lo siguiente para enumerar los archivos y el ID de memoria compartida:

```
sdlutil (logfile set) -files
sdlutil (logfile set) -id
```

Para suprimir los archivos, se utiliza el comando UNIX `rm nombre_de_archivo`. Para suprimir el ID de memoria compartida, se utiliza el comando UNIX `ipcrm -m id`. Obsérvese que sólo existe el ID de memoria compartida y habrá que eliminarlo si `sdlutil` no se ha eliminado correctamente cuando se cerró el conjunto de archivos de registro.

Mensaje SDL80

ERROR: Summarization metric (metric) not found in class.

No se definió anteriormente la métrica `SUMMARIZED BY` definida en la sección `METRIC`.

Acción: Consulte [Descripciones de métrica](#) en Capítulo 14 para obtener más información.

Mensaje SDL81

ERROR: Metric id (id) already defined in SDL.

El *id_de_métrica* sólo debe definirse una vez. Para reutilizar una definición de métrica que se haya definido anteriormente en otra clase, hay que especificar sólo el parámetro *nombre_de_métrica* sin *id_de_métrica* u otras especificaciones.

Acción: Consulte [MÉTRICA](#) en Capítulo 14 para obtener más información.

Mensaje SDL82

ERROR: Metric name (name) already defined in SDL.

El *nombre_de_métrica* sólo debe definirse una vez. Para reutilizar una definición de métrica que se haya definido anteriormente en otra clase, hay que especificar sólo el parámetro *nombre_de_métrica* sin *id_de_métrica* u otras especificaciones.

Acción: Consulte [MÉTRICA](#) en Capítulo 14 para obtener más información.

Mensaje SDL83

ERROR: Class id (id) already defined in SDL.

El *id_de_clase* sólo debe definirse una vez. Compruebe la ortografía para asegurarse de que se ha introducido correctamente.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL84

ERROR: Class name (name) already defined in SDL.

El *nombre_de_clase* sólo debe definirse una vez. Compruebe la ortografía para asegurarse de que se ha introducido correctamente.

Acción: Consulte [Sintaxis de especificación de clase](#) en Capítulo 14 para obtener más información.

Mensaje SDL85

ERROR: Must specify class to de-compile.

Hay que especificar una *lista_de_clase* al utilizar `-decomp`.

Acción: Consulte [Administrar datos con `sdlutil`](#) en Capítulo 15 para obtener más información.

Mensaje SDL87

ERROR: You must specify maximum number of classes with `-maxclass`.

Cuando se utiliza la opción `-maxclass`, hay que especificar el número de clases máximo que se va a proporcionar al crear un nuevo conjunto de archivos de registro.

Acción: Consulte [Compilador `sdlcomp`](#) en Capítulo 15 para obtener más información.

Mensaje SDL88

ERROR: Option \!"\" is not valid.

La entrada de línea de comandos no es válida.

Acción: Comprobar lo que se ha introducido para asegurarse de que sigue la sintaxis correcta.

Mensaje SDL89

ERROR: Maximum number of classes (!) for -maxclass is not valid.

El número -maxclass debe ser mayor que cero.

Acción: Consulte [Compilador sdlcomp](#) en Capítulo 15 para obtener más información.

Mensaje SDL90

ERROR: -f option but no result file specified.

Hay que especificar un archivo de formato al usar la opción -f.

Acción: Hay que especificar un archivo de formato al usar la opción -f.

Mensaje SDL91

ERROR: No specification file named.

No se ha asignado ningún nombre al archivo de especificación de clase.

Acción: Hay que introducir un *archivo_de_especificación* al utilizar `sdlcomp`. Consulte [Compilador sdlcomp](#) en Capítulo 15 para obtener más información.

Mensaje SDL92

ERROR: No log file set named.

Hay que introducir un *conjunto_de_archivos_de_registro* al utilizar `sdlcomp`.

Acción: Consulte [Compilador sdlcomp](#) en Capítulo 15 para obtener más información.

Mensaje SDL93

ERROR: Metric ID already defined in class.

El *id_de_métrica* sólo debe definirse una vez.

Acción: Para reutilizar una definición de métrica que se haya definido anteriormente en otra clase, hay que especificar sólo el parámetro *nombre_de_métrica* sin *id_de_métrica* u otras especificaciones.

Consulte [Descripciones de métrica](#) en Capítulo 14 para obtener más información.

Mensaje SDL94

ERROR: Metric name already defined in class.

El nombre de métrica sólo debe definirse una vez.

Acción: Para reutilizar una definición de métrica que se haya definido anteriormente en otra clase, hay que especificar sólo el parámetro *nombre_de_métrica* sin *id_de_métrica* u otras especificaciones. Consulte [Descripciones de métrica](#) en Capítulo 14 para obtener más información.

Mensaje SDL95

ERROR: Text found after complete class specification.

El compilador `sdlcomp` ha encontrado texto que no reconoció como parte de la especificación de clase.

Acción: Hay que volver a introducir la especificación y volver a intentarlo.

Mensaje SDL96

ERROR: Collection rate statement not valid.

El formato correcto es `RECORDS PER HOUR (número)`. Las palabras clave deben estar presentes en este orden y no se pueden abreviar.

Acción: Hay que corregir la palabra clave y seguir el formato requerido.

Mensaje SDL97

ERROR: Expecting integer between 1 and 2,147,483,647.

Hay que utilizar un número en este rango.

Acción: Hay que introducir un número que se encuentre en este rango.

Mensaje SDL98

ERROR: Action requires preceding ROLL BY statement.

La entrada está fuera de orden o falta en el archivo de especificación de clase.

Acción: La acción especifica lo que ocurrirá cuando el archivo de registro se suprime. Es importante saber primero cuándo se debería suprimir. `ROLL BY` debe preceder a `ACTION`.

Por ejemplo:

```
class xxxxxx = 10
  index by month max indexes 12
  action "11 *";
```

debería haber sido:

```
class xxxxxx = 10
  index by month max indexes 12
  roll by month
  action "11 *";
```

Mensaje SDL99

ERROR: MAX INDEXES requiere que vaya precedido por la instrucción INDEX BY.

La entrada está fuera de orden o falta en el archivo de especificación de clase.

Acción: Para especificar un número máximo de índices, el programa tiene que saber dónde se está realizando un proceso `index by`. La instrucción `INDEX BY` debe preceder a `MAX INDEXES`.

Por ejemplo:

```
class xxxxx = 10
  max indexes 12
  label "this is a test";
```

debería haber sido:

```
class xxxxx = 10
  index by month
  max indexes 12
  label "this is a test";
```

Mensaje SDL100

WARNING: CAPACITY UNLIMITED not implemented, derived value used. (SDL-100)

Mensaje SDL101

ERROR: Derived capacity too large. (SDL-101)

Mensaje SDL102

ERROR: Text Length should not exceed 4096.

La longitud máxima permitida para la métrica de texto es 4096.

Mensaje SDL103

ERROR: RECORDS PER HOUR should not be greater than 3600 for logging summarized data.

Acción: RECORDS PER HOUR puede ser mayor que 3600 sólo para datos no resumidos. Hay que utilizar la opción `-u` para compilar.

Mensajes de error DSILOG

Los mensajes de error DSILOG son mensajes del proceso de registro de `dsilog` con el formato `DSILOG<número_de_mensajes>`.

Mensaje DSILOG1

ERROR: Self describing log file not specified.

Acción: Hay que corregir la línea de comandos y volver a intentarlo.

Mensaje DSILOG2

ERROR: Data class name not specified.

La clase de datos debe ser el segundo parámetro pasado a `dsilog`.

Acción: Hay que corregir la línea de comandos y volver a intentarlo.

Mensaje DSILog3

ERROR: Could not open data input file (name).

No se pudo abrir el archivo especificado en la línea de comandos. Aparece un error del sistema de archivos UNIX en la línea siguiente del mensaje de error.

Mensaje DSILog4

ERROR: OpenClass ("name") failed.

No se pudo abrir la clase especificada. Es posible que no se encuentre en el conjunto de archivos de registro o que el archivo de datos no sea accesible.

Acción: A continuación se muestran mensajes descriptivos que ofrecen una descripción del error interno o un error del sistema de archivos.

Mensaje DSILog5

ERROR: Open of root log file (name) failed.

No se pudo abrir el archivo raíz del conjunto de archivos de registro. El motivo se muestra en los mensajes descriptivos.

Mensaje DSILog6

ERROR: Time stamp not defined in data class.

Se creó la clase pero no se incluyó ninguna marca de hora.

Acción: Hay que utilizar `sdlutil` para suprimir el conjunto de archivos de registro y comenzar de nuevo.

Mensaje DSILog7

ERROR: (Internal error) AddPoint () failed.

`dsilog` intentó escribir un registro en el archivo de datos, pero no pudo. A continuación se muestran mensajes descriptivos.

Mensaje DSILog8

ERROR: Invalid command line parameter (name).

El parámetro mostrado no se reconoció como una opción de línea de comandos válida o estaba fuera de lugar en la línea de comandos.

Acción: Hay que corregir el parámetro de línea de comandos y volver a intentarlo.

Mensaje DSILog9

ERROR: Could not open format file (name).

No se pudo encontrar el archivo que dirige la correspondencia de las métricas entrantes con la clase de datos, o está inaccesible. A continuación se muestran mensajes descriptivos con el error del sistema de archivos UNIX.

Acción: Comprobar el archivo de especificación de clase para comprobar que está presente.

Mensaje DSILog10

ERROR: Illegal metric name (name).

El archivo de formato contiene un nombre de métrica superior al tamaño máximo del nombre de métrica o, por otro lado, no es conforme con la sintaxis de nombres de métricas.

Acción: Hay que corregir el nombre de métrica en la especificación de clase y volver a ejecutar dsilog.

Mensaje DSILog11

ERROR: Too many input metrics defined. Max 100.

Sólo se pueden especificar 100 métricas en el archivo de formato.

Acción: La entrada debería volverse a formatear externamente para dsilog, o el origen de datos debería dividirse en dos o más orígenes de datos.

Mensaje DSILog12

ERROR: Could not find metric (name) in class.

El nombre de métrica encontrado en el archivo de formato no se pudo encontrar en la clase de datos.

Acción: Hay que realizar las correcciones pertinentes y volver a intentarlo.

Mensaje DSILog13

ERROR: Required time stamp not found in input specification.

Se utilizó la opción de línea de comandos `-timestamp`, pero el archivo de formato no especificó dónde se podía encontrar la marca de hora en los datos entrantes.

Acción: Hay que especificar dónde se puede encontrar la marca de hora.

Mensaje DSILog14

ERROR: (number) errors, collection aborted.

Se detectaron errores graves al configurar la recopilación.

Acción: Hay que corregir los errores y volver a intentarlo. También se pueden usar las opciones `-vi` y `-vo` para comprobar los datos tal como llegan y como deberían registrarse.

Mensaje DSILog15

ERROR: Self describing log file and data class not specified.

La línea de comandos debe especificar el conjunto de archivos de registro y la clase de datos en donde registrar los datos.

Acción: Hay que corregir la entrada de línea de comandos y volver a intentarlo.

Mensaje DSILog16

ERROR: Self describing log file set root file (name) could not be accessed.
error=(number).

No se pudo abrir el archivo raíz del conjunto de archivos de registro.

Acción: Comprobar los mensajes descriptivos que van a continuación del problema.

Mensaje (sin numerar)

```
Metric null has invalid data  
Ignore to end of line, metric value exceeds maximum
```

Este mensaje de advertencia aparece cuando `dsilog` no registra ningún dato para una línea de entrada determinada. Esto ocurre cuando la entrada no se adapta al formato esperado por los archivos de registro DSI, como cuando están presentes en la entrada líneas en blanco o de encabezado, o cuando un valor de métrica excede la precisión especificada. En este caso, las líneas incorrectas se omitirán (no se registrarán). `dsilog` reanudará el registro de datos en la siguiente línea entrada válida.

Mensaje DSILog17

```
ERROR: Logfile set is created to log unsummarized data, could not log  
summarized data.
```

Acción: Si se ha creado el conjunto de archivos de registro usando la opción `-u` durante la compilación, hay que utilizar la opción `-s 0` para el registro usando `dsilog`. El uso de esta opción indica que los datos registrados no se resumen.

Mensajes de error generales

Error	Explicación
-3	Se ha intentado agregar más clases que las permitidas por <code>max-class</code> .
-5	No se pudo abrir el archivo que contiene los datos de clase.
-6	No se pudo leer el archivo.
-7	No se pudo escribir en el archivo.
-9	Se intentó escribir en el archivo de registro cuando no se solicitó el acceso de escritura.
-11	No se pudo encontrar el puntero a la clase.
-13	Archivo o estructura de datos no inicializados.
-14	No se pudo leer el archivo de descripción de clase
-15	No se pudo escribir en el archivo de descripción de clase
-16	No se encontraron todas las métricas necesarias para definir una clase en la clase de descripción de la métrica.
-17	El nombre de ruta de un archivo en el conjunto de archivos de registro tiene más de 1.024 caracteres
-18	El nombre de clase tiene más de 20 caracteres.
-19	El archivo no es un archivo raíz del conjunto de archivos de registro.
-20	El archivo no forma parte de un conjunto de archivos <code>lod</code> .
-21	El software actual no puede acceder al conjunto de archivos de registro.
-22	No se pudo obtener el segmento o ID de memoria compartida.
-23	No se pudo adjuntar al segmento de memoria compartida.
-24	No se pudo abrir el conjunto de archivos de registro.
-25	No se pudo determinar el directorio de trabajo actual.
-26	No se pudo leer el encabezado de clase desde el archivo de datos de clase.
-27	Error al abrir el archivo en el conjunto de archivos de registro.
-28	No se pudo abrir la clase de datos.
-29	Error de <code>Lseek</code> .
-30	No se pudo leer desde el archivo de registro.

Error	Explicación
-31	No se pudo escribir en el archivo de registro.
-32	Error al suprimir.
-33	Error de shmctl (REM_ID).
-34	El conjunto de archivos de registro está incompleto: falta el archivo raíz o de descripción.
-35	El archivo de registro de destino para agregar una clase no está en el conjunto de archivos de registro actual.

18 Descripción del seguimiento de transacciones

En este capítulo se describe lo siguiente:

- Mejora de la administración del rendimiento
- Un escenario: Procesamiento de pedidos en tiempo real
- Monitorización de datos de transacciones

Mejora de la administración del rendimiento

Puede mejorar su capacidad de administrar el rendimiento del sistema con la posibilidad de realizar un seguimiento de las transacciones de HP Operations Agent y HP GlancePlus.

Como el número de aplicaciones comerciales críticas distribuidas aumenta, los jefes de aplicaciones y del sistema necesitan más información para indicarles cómo está funcionando su tecnología de la información (TI) distribuida.

- ¿Ha dejado de responder su aplicación?
- ¿El tiempo de respuesta de la aplicación no es aceptable?
- ¿Se están cumpliendo sus objetivos de nivel de servicio (SLO)?

Las capacidades de seguimiento de las transacciones de Componente Performance Collection y GlancePlus permiten que los administradores de TI incorporen la capacidad en administración de extremo a extremo de su entorno TI cliente/servidor en términos de transacciones comerciales. Con Componente Performance Collection, puede definir lo que es una transacción comercial y capturar los datos de las transacciones que tengan sentido en el contexto de *su* negocio.

Cuando se instrumentan las aplicaciones con las llamadas a la API de ARM (Application Response Measurement) estándar, estos productos proporcionan unas amplias capacidades de seguimiento de transacciones y administración de extremo a extremo a través de plataformas de varios proveedores.

Ventajas del seguimiento de transacciones

- Proporciona una vista del cliente del tiempo transcurrido desde el principio hasta el final de una transacción.
- Proporciona datos de la transacción.
- Ayuda a administrar los contratos de nivel de servicio (SLA).

Estos temas se tratan más detalladamente en el resto de esta sección.

Vista del cliente de los tiempos de las transacciones

El seguimiento de las transacciones proporciona una vista del cliente del tiempo transcurrido desde el principio hasta el final de una transacción. Cuando se usa el seguimiento de las transacciones en un entorno de tecnología de información (TI), verá las siguientes ventajas:

- Puede realizar un seguimiento preciso del número de veces que se ejecuta una transacción.
- Puede ver cuánto tarda una transacción en completarse, en lugar de calcular el tiempo aproximado como ocurre ahora.
- Puede correlacionar los tiempos de las transacciones con la utilización de recursos del sistema.
- Puede usar sus propios datos de producción comercial en aplicaciones de administración del sistema, como los datos usados para la planificación de la capacidad, la administración del rendimiento, la contabilidad y las anulaciones.
- Puede lograr la optimización de las aplicaciones y una solución de problemas de rendimiento detallada en función de una unidad de trabajo real (su transacción), en lugar de representar trabajo real con definiciones abstractas del sistema y recursos del sistema.

Datos de transacciones

Cuando se han insertado llamadas a la API de ARM (Application Response Measurement) en una aplicación para marcar el inicio y el fin de cada transacción comercial, puede usar las siguientes herramientas de monitorización del rendimiento y los recursos para monitorizar los datos de las transacciones:

- Componente Performance Collection proporciona la funcionalidad de registro necesaria para registrar, generar informes y detectar alarmas en los datos de transacciones. Los datos de transacciones pueden verse en Performance Manager, Glance o exportando los datos desde los archivos de registro de Componente Performance Collection a los archivos a los que se puede acceder mediante hojas de cálculo y otras herramientas de generación de informes.
- Performance Manager crea gráficos de datos de rendimiento para solucionar problemas a corto plazo y para examinar tendencias y realizar análisis a largo plazo.
- Glance muestra datos detallados en tiempo real para monitorizar los sistemas y las transacciones minuto a minuto.
- Performance Manager, Glance o el explorador de mensajes de HP Operations Manager permiten monitorizar alarmas de cumplimiento del nivel de servicio.

Las métricas de transacciones específicas se describen en el [Capítulo 22, Métricas de transacciones](#)

Objetivos de nivel de servicio

Los objetivos de nivel de servicio (SLO) se derivan de los niveles de servicio establecidos que necesitan los usuarios de las aplicaciones empresariales. Normalmente los SLO se basan en el desarrollo del contrato de nivel de servicio (SLA). De los SLO proceden las métricas reales que los jefes de recursos de informática necesitan recopilar, monitorizar y almacenar y de las que necesitan generar informes para determinar si se están cumpliendo los niveles de servicio acordados para el usuario de las aplicaciones empresariales.

Un SLO puede ser tan simple como monitorizar el tiempo de respuesta de una sola transacción, o bien tan complejo como realizar un seguimiento de la disponibilidad del sistema.

Un escenario: Procesamiento de pedidos en tiempo real

Imagine un canal de compras en televisión que tiene mucho éxito y emplea a cientos de operadores telefónicos que toman los pedidos de los espectadores de distintos tipos de mercancía. Suponga que esta empresa usa un programa informático para introducir la información de los pedidos, comprobar la disponibilidad de la mercancía y actualizar el inventario. Podemos usar esta empresa ficticia para ilustrar cómo el seguimiento de las transacciones puede ayudar a que una organización cumpla los compromisos con los clientes y los SLO.

En función de las tareas críticas, el factor de satisfacción de los clientes, el factor de productividad y el tiempo máximo de respuesta, los jefes de recursos pueden determinar el nivel de servicio que desean proporcionar a sus clientes.

El [Capítulo 23, Ejemplos de seguimiento de transacciones](#) contiene un ejemplo de pseudocódigo de cómo las llamadas a la API de ARM pueden insertarse en una aplicación de procesamiento de pedidos de ejemplo, de forma que los datos de las transacciones puedan monitorizarse con Componente Performance Collection y Glance.

Requisitos del procesamiento de pedidos en tiempo real

Para cumplir los SLO en el ejemplo de procesamiento de pedidos en tiempo real antes descrito, los jefes de recursos deben realizar un seguimiento del tiempo necesario para completar las siguientes tareas críticas:

- Introducir la información del pedido
- Consultar la disponibilidad de la mercancía
- Actualizar el inventario

El principal factor de satisfacción del cliente es la rapidez con la que los operadores pueden tomar su pedido.

El principal factor de productividad para la empresa es el número de pedidos que los operadores pueden completar cada hora.

Para cumplir los factores de satisfacción del cliente y productividad, los tiempos de respuesta de las transacciones que acceden a la base de datos del inventario, ajustan el inventario y vuelven a escribir el registro deben monitorizarse para cumplir los SLO establecidos. Por ejemplo, los jefes de recursos pueden haber establecido un SLO para esta aplicación en el que se indique que el 90 por ciento de las transacciones deben completarse en cinco segundos o menos.

Preparación de la aplicación de procesamiento de pedidos

Las llamadas a la API de ARM pueden insertarse en aplicaciones de procesamiento de pedidos para crear transacciones para respuesta del inventario y actualizar inventario. Tenga en cuenta que las llamadas a la API de ARM deben insertarlas los programadores de aplicaciones *antes* de compilar la aplicación. Consulte el [Capítulo 23, Ejemplos de seguimiento de transacciones](#) para ver un programa de procesamiento de pedidos (escrito en pseudocódigo) que incluye llamadas a la API de ARM que definen diversas transacciones.

Para tener más información acerca de la instrumentación de aplicaciones con llamadas a la API de ARM, consulte *Application Response Measurement 2.0 API Guide* (Guía de la API de Application Response Measurement 2.0).

Monitorización de datos de transacciones

Cuando se instala y ejecuta en el sistema una aplicación que está instrumentada con llamadas a la API de ARM, puede monitorizar los datos de las transacciones con Componente Performance Collection, GlancePlus o Performance Manager.

... con Componente Performance Collection

Con Componente Performance Collection, puede recopilar y registrar datos para transacciones con nombres, monitorizar tendencias en sus SLO a lo largo del tiempo y generar alarmas cuando se superen los SLO. Una vez identificadas estas tendencias, los costes de informática pueden asignarse según el volumen de transacciones. Las alarmas de Componente Performance Collection pueden configurarse para activar el buscaperonas de un técnico, de forma que los problemas puedan investigarse y resolverse inmediatamente. Para obtener más información, consulte [Capítulo 24, Funciones avanzadas](#).

Componente Performance Collection es necesario para ver los datos de transacciones en Performance Manager.

... con Performance Manager

Performance Manager recibe alarmas y datos de transacciones de Componente Performance Collection. Por ejemplo, puede configurar Componente Performance Collection de forma que cuando una aplicación de procesamiento de pedidos tarde demasiado en comprobar el stock, Performance Manager reciba una alarma y envíe una advertencia a la consola del jefe de recursos para avisarle de un posible problema.

En Performance Manager, puede seleccionar **TRANSACTION** en la ventana Class List para un origen de datos y, a continuación, **graph transaction metrics** para varias transacciones. Para obtener más información, consulte la ayuda en línea de Performance Manager.

... con GlancePlus

Use GlancePlus para monitorizar el tiempo de respuesta de las transacciones al segundo y si las transacciones se están realizando de acuerdo con sus SLO establecidos. GlancePlus le ayuda a identificar y resolver los cuellos de botella en los recursos que puedan afectar al rendimiento de las transacciones. Para obtener más información, consulte la ayuda en línea de GlancePlus, a la que puede acceder mediante el menú Help de GlancePlus.

Directrices para usar ARM

La instrumentación de aplicaciones con la API de ARM requiere una cuidadosa planificación. Además, administrar el entorno que tiene aplicaciones sobre plataformas ARM es más fácil si se entienden las características y limitaciones de la recopilación de datos de ARM. A continuación se proporciona una lista de las áreas que podrían causar cierta confusión si no se entienden perfectamente.

- 1 Para capturar la métrica de ARM, `ttd` y `midaemon` debe estar ejecutándose. Para Componente Performance Collection, el compilador `scope` debe estar ejecutándose para registrar la métrica de ARM. La secuencia de comandos `ovpa start` inicia todos los procesos necesarios. De igual forma, Glance inicia `ttd` y `midaemon` si aún no están activos. (Consulte [Demonio de seguimiento de transacciones \(ttd\)](#) en Capítulo 19).
- 2 Volver a leer el archivo de configuración de transacciones, `ttd.conf`, para capturar cualquier nombre de transacción recién definido. (Consulte [Archivo de configuración de transacciones \(ttd.conf\)](#) en Capítulo 19).
- 3 Componente Performance Collection, las aplicaciones de usuario y `ttd` deben reiniciarse para capturar cualquier rango de transacciones y objetivos de nivel de servicio (SLO) *nuevos* o modificados. (Consulte [Adición de nuevas aplicaciones](#) en Capítulo 19).
- 4 Componente Performance Collection omite las cadenas en la métrica definida por el usuario. Sólo se registran las seis primeras métricas definidas por el usuario que no sean cadenas. (Consulte [Cómo se usan los tipos de datos](#) en Capítulo 24).
- 5 El uso de guiones en el nombre de las transacciones tiene limitaciones si está especificando una condición de alarma para esa transacción. (Consulte “... con Componente Performance Collection” en la sección [Alarmas](#) en Capítulo 20).
- 6 Componente Performance Collection sólo mostrará los 60 primeros caracteres del nombre de la aplicación y el nombre de la transacción. (Consulte [Especificación de nombres de aplicaciones y transacciones](#) en Capítulo 19).
- 7 Limite el número de nombres de transacción exclusivos que se instrumentan. (Consulte [Límites en transacciones exclusivas](#) en Capítulo 20).
- 8 No permita que las llamadas a la función API de ARM afecten a la ejecución de una aplicación desde la perspectiva de un usuario final. (Consulte [Devolución del estado una llamada a la API de ARM](#) en Capítulo 19).
- 9 Use bibliotecas compartidas para la vinculación. (Consulte la sección [Ejemplos de opción de compilador C por plataforma](#) en la página 394).

19 Funcionamiento del seguimiento de transacciones

Los siguientes componentes de Componente Performance Collection y GlancePlus funcionan juntos para ayudarle a definir y realizar un seguimiento de los datos de transacciones procedentes de aplicaciones instrumentadas con llamadas a Application Response Measurement (ARM).

- El demonio de la interfaz de medición, `midaemon`, es un proceso de demonio que monitoriza en forma de datos de transacciones en su segmento de memoria compartida, donde Componente Performance Collection, Performance Manager y GlancePlus pueden acceder a la información y generar informes. En sistemas HP-UX, `midaemon` también monitoriza los datos de rendimiento del sistema.
- El archivo de configuración de las transacciones, `/var/opt/perf/ttd.conf`, se usa para definir las transacciones e identificar la información que se monitoriza para cada transacción.
- El demonio de seguimiento de transacciones, `ttd`, lee, registra y sincroniza las definiciones de transacciones desde el siguiente archivo de configuración, `ttd.conf`, con `midaemon`.

Compatibilidad con ARM 2.0

ARM 2.0 es un superconjunto de la versión anterior de Application Response Measurement. Las nuevas funciones que ARM 2.0 proporciona son métricas definidas por el usuario, correlación de transacciones y un agente de registro. Componente Performance Collection y GlancePlus son compatibles con métricas definidas por el usuario y la correlación de transacciones, pero *no* son compatibles con el agente de registro.

Sin embargo, puede desear usar el agente de registro para probar la instrumentación en su aplicación. El código de origen del agente de registro, `logagent.c`, está incluido en el kit de desarrollo de software (SDK) de ARM 2.0 que está disponible en el siguiente sitio web:

<http://regions.cmg.org/regions/cmgarmlw>

Para obtener información acerca del uso del agente de registro, consulte *Application Response Measurement 2.0 API Guide* (Guía de la API de Application Response Measurement 2.0).



En el manual *Application Response Measurement 2.0 API Guide* (Guía de la API de Application Response Measurement 2.0) se usa el término “métrica definida por la aplicación” en lugar de “métrica definida por el usuario”.

Compatibilidad con llamadas a la API de ARM

Las llamadas a la API de Application Response Measurement (ARM) enumeradas a continuación pueden usarse en Componente Performance Collection y GlancePlus.

<code>arm_init()</code>	Asigna un nombre al usuario y registra la aplicación y (opcionalmente) el usuario.
<code>arm_getid()</code>	Asigna un nombre al usuario y registra una clase de transacción y proporciona información relacionada con la transacción. Define el contexto de la métrica definida por el usuario.
<code>arm_start()</code>	Señala el inicio de un caso de transacción exclusiva.
<code>arm_update()</code>	Actualiza los valores de un caso de transacción exclusiva.
<code>arm_stop()</code>	Señala el final de un caso de transacción exclusiva.
<code>arm_end()</code>	Señala el final de la aplicación.

Consulte el manual *Application Response Measurement 2.0 API Guide* (Guía de la API de Application Response Measurement 2.0) actual y la página del manual *arm* (3) para obtener información acerca de la instrumentación de aplicaciones con llamadas a la API de ARM, así como completar descripciones de las llamadas y sus parámetros. Para aplicaciones comerciales, compruebe la localización del producto para ver si la aplicación ha sido instrumentada con llamadas a la API de ARM.

Para obtener información importante acerca de las bibliotecas necesarias, consulte [Bibliotecas de transacciones](#) en la página 387, más adelante en este manual.

Llamada `arm_complete_transaction`

Además de la API de ARM 2.0 estándar, el agente HP ARM admite el uso de la llamada `arm_complete_transaction`. Esta llamada, que es una extensión específica de HP de la ARM estándar, puede usarse para marcar el final de una transacción que se ha completado cuando no se pudo delimitar el inicio de la transacción mediante una llamada `arm_start`. La llamada `arm_complete_transaction` toma como parámetro el tiempo de respuesta del ejemplo de transacción completada.

Además de marcar el final de una instancia de transacción, puede proporcionarse información adicional sobre la transacción en el búfer de datos opcional. Consulte la página del manual *arm (3)* para obtener más información acerca de estos datos opcionales, así como una descripción completa de esta llamada y sus parámetros.

Ejemplo de aplicaciones instrumentadas para ARM

Para ver ejemplos de cómo se implementan las llamadas a la API de ARM, consulte el ejemplo de las aplicaciones instrumentadas para ARM, `armsample1.c`, `armsample2.c`, `armsample3.c` y `armsample4.c` y la secuencia de comandos incorporada, `Make.armsample`, en el directorio `<dir_instal>/examples/arm/`.

- `armsample1.c` muestra el uso de llamadas a la API de ARM estándar simples.
- `armsample2.c` también muestra el uso de llamadas a la API de ARM estándar simples. Su estructura es similar a `armsample1.c`, pero es interactiva.
- `armsample3.c` proporciona ejemplos de cómo usar la métrica definida por el usuario y el correlador de transacciones, proporcionados por la versión 2.0 de la API de ARM. Este ejemplo simula una aplicación cliente/servidor en la que tanto el cliente como el servidor realizan una serie de transacciones. (Normalmente los componentes cliente y servidor de la aplicación existirían en programas independientes, pero se han colocado juntos por sencillez).

El procedimiento del cliente inicia una traslación y solicita un correlador de ARM de su llamada a `arm_start`. El cliente guarda este correlador y la pasa al servidor para que éste pueda usarlo cuando llama a `arm_start`. A continuación, las herramientas de rendimiento que se ejecutan en el servidor pueden usar esta información del correlador para definir los distintos clientes que usan el servidor.

En este programa también se muestra el mecanismo para pasar valores de la métrica definida por el usuario a la API de ARM. Esto permite no sólo ver la información de los tiempos de respuesta y el nivel de servicio en las herramientas de rendimiento, sino también los datos que pueden ser importantes para la aplicación en sí. Por ejemplo, una transacción puede estar procesando solicitudes de distintos tamaños; y el tamaño de la solicitud puede ser una métrica definida por el usuario. Cuando los tiempos de respuesta son altos, esta métrica definida por el usuario puede usarse para ver si los tiempos de respuesta largos corresponden a transacciones con un mayor tamaño.

- `armsample4.c` proporciona un ejemplo de uso de las métricas definidas por usuario en llamadas a ARM. Se pueden pasar distintos valores de la métrica a través de las llamadas `arm_start`, `arm_update` y `arm_stop`. Si lo prefiere, se puede usar `arm_complete_transaction`, cuando `tran` no puede estar delimitado por las llamadas `start/stop`.

Especificación de nombres de aplicaciones y transacciones

Aunque ARM permite un máximo de 128 caracteres cada uno para nombres de transacciones y aplicaciones en las llamadas a la API `arm_init` y `arm_getid`, Componente Performance Collection *sólo* muestra un máximo de 60 caracteres. Los caracteres después de los primeros 60 no se verán. Sin embargo, GlancePlus permite ver hasta 128 caracteres.

Componente Performance Collection aplica ciertas limitaciones acerca de cómo los nombres de aplicaciones y transacciones se muestran en los datos de transacción extraídos o exportados. Estas normas también se usan para ver los nombres de aplicaciones y transacciones en Performance Manager.

El nombre de la aplicación *siempre* tiene prioridad sobre nombre de la transacción. Por ejemplo, si está exportando datos de la transacción que tienen un nombre de aplicación de 65 caracteres y un nombre de transacción de 40 caracteres, *sólo* se muestra el nombre de la aplicación. Los últimos cinco caracteres del nombre de la aplicación no se muestran.

Para otro ejemplo, si el nombre de la aplicación tiene 32 caracteres y el nombre de la transacción tiene 40 caracteres, Componente Performance Collection muestra el nombre de la aplicación completa, pero el nombre de la transacción aparece truncado. Se muestra un total de 60 caracteres. 59 caracteres se asignan a los nombres de la aplicación y la transacción y un carácter se asigna al signo de subrayado (`_`) que separa los dos nombres. El nombre de la aplicación “WarehouseInventoryApplication” y el nombre de la transacción “CallFromWestCoastElectronicSupplier” aparecerían en Componente Performance Collection o Performance Manager de la forma siguiente:

```
WarehouseInventoryApplication_CallFromWestCoastElectronicSup
```



La combinación de 60 caracteres del nombre de la aplicación y el nombre de la transacción debe ser exclusiva si los datos se van a ver con Performance Manager.

Demonio de seguimiento de transacciones (ttd)

El demonio de seguimiento de transacciones, `ttd`, lee, registra y sincroniza las definiciones de transacciones desde `ttd.conf` con `midaemon`.

`ttd` se inicia al iniciar el recopilador de datos `scope` de Componente Performance Collection con el comando `ovpa start.ttd` se ejecuta en segundo plano cuando se envían, y los errores se escriben en el archivo `/var/opt/perf/status.ttd`.

`midaemon` también debe estar ejecutándose para procesar las transacciones y recopilar las métricas de rendimiento asociadas a estas transacciones (consulte la página siguiente).



Recomendamos encarecidamente no detener `ttd`.

Si debe detener `ttd`, todas las aplicaciones instrumentadas por ARM que se están ejecutando *deben* detenerse también antes de reiniciar los procesos `ttd` y Componente Performance Collection. `ttd` debe estar ejecutándose para capturar todas las llamadas `arm_init` y `arm_getid` que se realizan en el sistema. Si se detiene y se reinicia `ttd`, los Id. de transacciones devueltos por estas llamadas se repetirán, lo que invalida la métrica de ARM

Use la secuencia de comandos `ovpa` para iniciar los procesos de Componente Performance Collection para garantizar que los procesos se inician en orden correcto. `ovpa stop no` cerrará `ttd`. Si `ttd` debe cerrarse para reinstalar un software de rendimiento, use el comando `<dir_instal>/bin/ttd -k`. Sin embargo, no recomendamos detener `ttd`, excepto al reinstalar Componente Performance Collection.

Si Componente Performance Collection no está en su sistema, GlancePlus inicia `midaemon`. A continuación, `midaemon` inicia `ttd` si no se está ejecutando *antes de que* `midaemon` empiece a procesar cualquier dato medido.

Consulte la página del manual de `ttd` para ver todas las opciones del programa.

Devolución del estado una llamada a la API de ARM

El proceso `ttd` siempre debe estar ejecutándose para registrar las transacciones. Si se termina `ttd` por cualquier razón, mientras no se está ejecutando, la llamadas a `arm_init` o `arm_getid` devolverán un código de retorno de error. Si `ttd` se reinicia después, las nuevas llamadas a `arm_getid` pueden volver a registrar los mismos Id. de las acciones que ya están en uso por otros programas, lo que hace que se registren datos no válidos.

Cuando se termina y reinicia `ttd`, las aplicaciones instrumentadas por ARM pueden empezar a tener un valor de retorno de `-2` (`TT_TTDNOTRUNNING`) y un error `EPIPE` `errno` en llamadas a la API de ARM. Cuando la aplicación se pone en marcha inicialmente, se crea un controlador de conexión del cliente en cualquier llamada a la API de ARM. Este controlador del cliente permite a la aplicación comunicarse con el proceso `ttd`. Cuando se termina `ttd`, esta conexión ya no es válida y la siguiente vez que la aplicación intenta usar una llamada a la API de ARM, puede obtener un valor de retorno de `TT_TTDNOTRUNNING`. Este error refleja que el *anterior* proceso `ttd` ya no se ejecuta, ni siquiera si hay otro proceso `ttd` ejecutándose. (Algunos de los resultados de las llamadas a la API de ARM se explican en la página del manual *arm* (3)).

Para evitar este problema, debe reiniciar las aplicaciones instrumentadas por ARM si se termina `ttd`. Primero, detenga las aplicaciones sobre plataformas ARM. A continuación, reinicie `ttd` (con `<dir_instal>/bin/ovpa start` o `<dir_instal>/bin/ttd`) y, a continuación, reinicie las aplicaciones. El reinicio de la aplicación provoca la creación de un controlador de conexión del cliente entre la aplicación y el proceso `ttd`.

Algunas llamadas a la API de ARM no devolverán ningún error si `midaemon` tiene un error. Por ejemplo, esto se produciría si `midaemon` se ha quedado sin espacio en su segmento de memoria compartida. La métrica del rendimiento `GBL_TT_OVERFLOW_COUNT` será > 0 . Si se produce una situación de desbordamiento, puede desear cerrar cualquier herramienta de rendimiento que se esté ejecutando (excepto `tttd`) y reiniciar `midaemon` con la opción `-smdvss` para especificar más espacio en el segmento de memoria compartida. (Para obtener más información, consulte la página del manual de *midaemon*).

Recomendamos que las aplicaciones se escriban de forma que continúen ejecutándose incluso si se produce algún error de ARM. El estado de ARM no debería afectar a la ejecución del programa.

El número de procesos de clientes activos que pueden registrar transacciones con `tttd` a través de la llamada `arm_getid` se limita al parámetro del núcleo `maxfiles`. Este parámetro controla el número de archivos abiertos por proceso. Cada solicitud de registro del cliente hace que `tttd` abra `socket` (un archivo abierto) para la conexión RPC. El `socket` se cierra cuando la aplicación cliente termina. Por tanto, este límite sólo afecta al número de clientes activos que han registrado una transacción mediante la llamada a `arm_getid`. Una vez alcanzado este límite, `tttd` devolverá `TT_TTDNOTRUNNING` a una solicitud `arm_getid` de un cliente. El parámetro del núcleo `maxfiles` puede aumentarse para elevar este límite por encima del número de aplicaciones activas que registrarán transacciones con `tttd`.

Demonio de interfaz de medición (*midaemon*)

El demonio de interfaz de medición, *midaemon*, es un proceso con poca sobrecarga que recopila continuamente información de rendimiento del sistema. *midaemon* debe estar ejecutándose para que Componente Performance Collection recopile datos de transacciones o para que GlancePlus genere informes de los datos de transacciones. Se pone en marcha cuando se ejecuta `scope` con el comando `ovpa start` o cuando se inicia GlancePlus.

Componente Performance Collection y GlancePlus requieren que *midaemon* y `tttd` estén ejecutándose para que estas transacciones puedan registrarse y rastrearse. La secuencia de comandos `ovpa` inicia y detiene el procesamiento de Componente Performance Collection, incluido *midaemon*, en el orden correcto. GlancePlus inicia *midaemon*, si aún no se está ejecutando. *midaemon* inicia `tttd`, si aún no se está ejecutando.

Consulte la sección [Sobrecargade la CPU](#) más adelante en este manual para obtener información acerca de la sobrecarga de la CPU de *midaemon*.

Consulte la página del manual de *midaemon* para ver todas las opciones del programa.

Archivo de configuración de transacciones (ttd.conf)

El archivo de configuración de transacciones, `/var/opt/perf/ttd.conf`, permite definir el nombre de la aplicación, el nombre de la transacción, los rangos de distribución del rendimiento y el objetivo de nivel de servicio que se desea cumplir en cada transacción. `ttd` lee `ttd.conf` para determinar cómo registrar cada transacción.

La personalización de `ttd.conf` es opcional. El archivo de configuración de transacciones que se proporciona con Componente Performance Collection hace que se monitoricen *todas* las transacciones instrumentadas en cualquier aplicación.

Si está usando una aplicación comercial y no sabe las transacciones que se han instrumentado en la aplicación, recopile algunos datos usando el archivo `ttd.conf` predeterminado. A continuación, mire los datos para ver las transacciones que están disponibles. A continuación, personalice la recopilación de datos de transacciones para esa aplicación modificando `ttd.conf`.

Adición de nuevas aplicaciones

Si agrega nuevas aplicaciones sobre plataformas ARM al sistema que utilizan los valores `slo` y `range` de la línea `tran=*` en el archivo `ttd.conf`, no hay que hacer nada para incorporar estas nuevas transacciones. (Consulte la sección [Palabras clave de archivos de configuración](#) para ver descripciones de `tran`, `range` y `slo`). Las nuevas transacciones se obtendrán automáticamente. Los valores de `slo` y `range` de la línea `tran` en el archivo `ttd.conf` se aplicará a las nuevas transacciones.

Adición de nuevas transacciones

Después de realizar adiciones en el archivo `ttd.conf`, debe llevar a cabo los siguientes pasos para que las adiciones entren en vigor:

- Detenga todas las aplicaciones.
- Ejecute el comando `ttd -hup -mi como raíz`.

Las anteriores acciones hacen que se vuelva a leer el archivo `ttd.conf` y registran las nuevas transacciones, junto con los valores de `slo` y `range` con `ttd` y `midaemon`. La nueva lectura no cambiará los valores de `slo` ni `range` para ninguna transacción que estaba en el archivo `ttd.conf` antes de la nueva lectura.

Cambio de los valores de range o SLO

Si necesita cambiar los valores de SLO o range para transacciones existentes en el archivo `ttd.conf`, haga lo siguiente:

- Detenga todas las aplicaciones sobre plataformas ARM.
- Detenga el recopilador `scope` mediante **`ovpa stop`**.
- Detenga el uso de Glance.
- Detenga `ttd` emitiendo el comando **`ttd -k`**.
- Una vez que haya realizado los cambios en el archivo `ttd.conf`:
- Reinicie `scope` con **`ovpa start`**.
- Reinicie las aplicaciones sobre plataformas ARM.

Palabras clave de archivos de configuración

El archivo de configuración `/var/opt/perf/ttd.conf` asocia los nombres de las transacciones a los atributos de las transacciones definidas por las palabras clave en la [Tabla 1](#)

Tabla 1 Palabras clave de archivos de configuración

Palabra clave	Sintaxis	Uso
<code>tran</code>	<code>tran=nombre_transacción</code>	Obligatorio
<code>slo</code>	<code>slo=seg.</code>	Opcional
<code>range</code>	<code>range=seg. [,seg,...]</code>	Opcional

Estas palabras clave se describen con más detalle a continuación.

tran

Use `tran` para definir el nombre de la transacción. Este nombre debe corresponder a una transacción definida en la llamada a la API `arm_getid` en su aplicación instrumentada. Debe usar la palabra clave `tran` antes de poder especificar los atributos opcionales `range` o `slo`. `tran` es la única palabra clave obligatoria en el archivo de configuración. Un asterisco (*) delante del nombre de la transacción hace que se realice una búsqueda del patrón de comodín cuando se realizan solicitudes de registro para esta entrada. Pueden usarse guiones en un nombre de transacción. Sin embargo, no pueden usarse espacios en un nombre de transacción.

El nombre de la transacción puede contener un máximo de 128 caracteres. Sin embargo, sólo los primeros 60 caracteres se verán en Componente Performance Collection. GlancePlus puede mostrar 128 caracteres en pantallas concretas.

El archivo `ttd.conf` predeterminado consta de varias entradas. Las primeras entradas definen las transacciones usadas por el recopilador de datos de Componente Performance Collection `scope`, que se ha instrumentado con llamadas a la API de ARM. El archivo también contiene la entrada `tran=*`, que registra todas las demás transacciones en aplicaciones instrumentadas con llamadas a la API de ARM o Transaction Tracker.

range

Use `range` para especificar los rangos de distribución del rendimiento de las transacciones. Los rangos de distribución del rendimiento permiten distinguir entre transacciones que tardan distintos tiempos en completarse y ver el número de transacciones correctas con cada tiempo. Los rangos que define aparecen en la ventana GlancePlus Transaction Tracking.

Cada valor introducido para `sec` representa el límite superior en segundos para el tiempo de la transacción para el rango. El valor puede ser un número entero o real con un máximo de seis dígitos a la derecha de la coma decimal. En HP-UX, esto permite una precisión de un microsegundo (0,000001 segundos). Sin embargo, en otras plataformas la precisión es de 10 milisegundos (0,01 segundos), por lo que sólo se reconocen los primeros dos dígitos situados a la derecha de la coma decimal.

Se admite un máximo de 10 rangos para cada transacción definida.

Puede especificar un máximo de nueve rangos. Se reserva un rango para un rango de desbordamiento, que recopila datos de las transacciones que tardan más que el rango más largo definido por el usuario. Si especifica más de nueve rangos, se usan los primeros nueve y los demás se ignoran.

Si especifica menos de nueve rangos, el primer rango no especificado pasa a ser el rango de desbordamiento. Los demás rangos no especificados no se utilizan. La métrica de los rangos no especificados aparece como 0,000. La primera métrica de recuento no especificada correspondiente pasa a ser el recuento de desbordamiento. Las demás métricas de recuento no especificadas aparece como cero (0).

Los rangos deben definirse en orden ascendente (consulte ejemplos más adelante en este capítulo).

slo

Use `slo` para especificar en segundos el objetivo de nivel de servicio (SLO) que desee usar para supervisar el contrato de nivel de servicio (SLA).

Al igual que con la palabra clave `range`, el valor puede ser un número entero o real con un máximo de seis dígitos a la derecha de la coma decimal. En HP-UX, esto permite una precisión de un microsegundo (0,000001 segundos). Sin embargo, en otras plataformas la precisión es de 10 milisegundos (0,01 segundos), por lo que sólo se reconocen los primeros dos dígitos situados a la derecha de la coma decimal.

Tenga en cuenta que aunque las transacciones puedan ordenarse con una precisión de 1 microsegundo en HP-UX, los tiempos de las transacciones aparecen con una precisión de 100 microsegundos.

Formato del archivo de configuración

El archivo `ttd.conf` puede contener dos tipos de entradas: transacciones generales y transacciones específicas de la aplicación.

Las transacciones generales deben definirse en el archivo `ttd.conf` antes de definir cualquier aplicación. Estas transacciones se asociarán a todas las aplicaciones que están definidas. El archivo predeterminado `ttd.conf` contiene una entrada de transacción general y entradas para el recopilador `scope` que se instrumenta con llamadas a la API de ARM.

```
tran=* range=0.5, 1, 2, 3, 5, 10, 30, 120, 300 slo=5.0
```

Opcionalmente, cada aplicación puede tener su propio conjunto de nombre de transacciones. Estas transacciones *sólo* se asociarán a esa aplicación. El nombre de la aplicación que especifique debe corresponder a un nombre de aplicación definido en la llamada a la API `arm_getid` en su aplicación instrumentada. Cada grupo de entradas específicas de la aplicación debe empezar por el nombre de la aplicación encerrado entre corchetes. Por ejemplo:

```
[AccountRec]
tran=acctOne range=0.01, 0.03, 0.05
```

El nombre de la aplicación puede contener un máximo de 128 caracteres. Sin embargo, sólo los primeros 60 caracteres se verán en Componente Performance Collection. Glance puede mostrar 128 caracteres en pantallas concretas.

Si hay transacciones que tienen el mismo nombre que una transacción “general”, se usará la transacción enumerada bajo la aplicación.

Por ejemplo:

```
tran=abc range=0.01, 0.03, 0.05 slo=0.10
tran=xyz range=0.02, 0.04, 0.06 slo=0.08
tran=t* range=0.01, 0.02, 0.03

[AccountRec]
tran=acctOne range=0.04, 0.06, 0.08
tran=acctTwo range=0.1, 0.2
tran=t* range=0.03, 0.5

[AccountPay]

[GenLedg]
tran=GenLedgOne range=0.01
```

En el ejemplo anterior, las tres primeras transacciones se aplican a las tres aplicaciones especificadas.

La aplicación `[AccountRec]` tiene las siguientes transacciones: `acctOne`, `acctTwo`, `abc`, `xyz` y `t*`. Una de las entradas del conjunto de transacciones generales también tiene una transacción con comodín denominada `"t*"`. En este caso, se usará el nombre de la transacción `"t*"` de la aplicación `AccountRec`; el del conjunto de transacciones generales se ignorará.

La aplicación `[AccountPay]` sólo tienen transacciones del conjunto de transacciones generales.

La aplicación `[GenLedg]` tiene transacciones `GenLedgOne`, `abc`, `xyz` y `t*`.

El orden de los nombres de las transacciones no representa ninguna diferencia en la aplicación.

Para obtener información adicional acerca de los nombres de las aplicaciones y las transacciones, consulte la sección [Especificación de nombres de aplicaciones y transacciones](#) en este capítulo.

Ejemplos de archivos de configuración

Ejemplo 1

```
tran=* range=0.5,1,2,3,5,10,30,12,30 slo=5.0
```

La entrada "*" se usa como valor predeterminado si ninguna de las entradas coincide con un nombre de transacción registrado. Estos valores predeterminados pueden cambiarse en cada sistema modificando la entrada "*". Si falta la entrada "*", se usa un conjunto predeterminado de parámetros de registro que coinciden con los parámetros iniciales asignados a la anterior entrada "*".

Ejemplo 2

```
[MANufactur]
tran=MFG01 range=1,2,3,4,5,10 slo=3.0
tran=MFG02 range=1,2.2,3.3,4.0,5.5,10 slo=4.5
tran=MFG03
tran=MFG04 range=1,2.2,3.3,4.0,5.5,10
```

Las transacciones de la aplicación MANufactur, MFG01, MFG02 y MFG04, usan sus propios parámetros exclusivos. La transacción MFG03 no necesita realizar un seguimiento de las distribuciones de tiempo o los objetivos de nivel de servicio, por lo que no especifica estos parámetros.

Ejemplo 3

```
[Financial]
tran=FIN01
tran=FIN02 range=0.1,0.5,1,2,3,4,5,10,20 slo=1.0
tran=FIN03 range=0.1,0.5,1,2,3,4,5,10,20 slo=2.0
```

Las transacciones de la aplicación Financial, FIN02 y FIN03, usan sus propios parámetros exclusivos. La transacción FIN01 no necesita realizar un seguimiento de las distribuciones de tiempo o los objetivos de nivel de servicio, por lo que no especifica estos parámetros.

Ejemplo 4

```
[PERSONL]
tran=PERS* range=0.1,0.5,1,2,3,4,5,10,20 slo=1.0
tran=PERS03 range=0.1,0.2,0.5,1,2,3,4,5,10,20 slo=0.8
```

La transacción PERS03 de la aplicación PERSONL usa sus propios parámetros exclusivos, mientras que el resto de las transacciones de personal usa el conjunto de parámetros predeterminado exclusivo de la aplicación PERSONL.

Ejemplo 5

```
[ACCOUNTS]
tran=ACCT_* slo=1.0
tran=ACCT_REC range=0.5,1,2,3,4,5,10,20 slo=2.0
tran=ACCT_PAY range=0.5,1,2,3,4,5,10,20 slo=2.0
```

Las transacciones de la aplicación ACCOUNTS, ACCT_REC y ACCT_PAY, usan sus propios parámetros exclusivos, mientras que el resto de las transacciones de contabilidad usa el conjunto de parámetros predeterminado exclusivo de la aplicación de contabilidad. Sólo las transacciones de cuentas a pagar y cuentas a cobrar deben realizar un seguimiento de las distribuciones de tiempo. El orden de los nombres de las transacciones no representa ninguna diferencia en la aplicación.

Consideraciones de sobrecarga para usar ARM

Las versiones actuales de Componente Performance Collection y GlancePlus contienen modificaciones realizadas en la interfaz de medición que admiten datos adicionales necesarios para ARM 2.0. Estas modificaciones pueden dar como resultado una mayor sobrecarga en la administración del rendimiento. Debe tener en cuenta las consideraciones de sobrecarga al planificar la instrumentación de ARM para sus aplicaciones.

Las áreas de sobrecarga que tratan en el resto de este capítulo.

Directrices

A continuación se proporcionan algunas directrices que se deben seguir al instrumentar las aplicaciones con la API de ARM:

- El número total de Id. de transacciones independientes debe estar limitado a no más de 4.000. Normalmente, es más barato tener varios casos de la misma transacción que tener un solo caso de cada transacción. Registre *sólo* las transacciones que se monitorizarán activamente.
- Aunque la sobrecarga de las llamadas a la API `arm_start` y `arm_stop` es muy pequeña, puede aumentar si hay un gran volumen de casos de transacciones. Más de unos pocos miles de llamadas `arm_start` y `arm_stop` por segundo en la mayoría de los sistemas puede tener un impacto importante en el rendimiento global.
- Solicite correladores de ARM *sólo* cuando se use la funcionalidad ARM 2.0. (Para obtener información acerca de los correladores de ARM, consulte la sección de temas avanzados en *Application Response Measurement 2.0 API Guide* (Guía de la API de Application Response Measurement 2.0). La sobrecarga de producir, mover y monitorizar la información del correlador es mucho más alta que la de monitorizar transacciones no instrumentadas para usar la funcionalidad del correlador de ARM 2.0.
- Un tamaño de cadena mayor (aplicaciones que registran nombres de transacciones, nombres de aplicaciones y métricas de cadenas definidas por el usuario largos) supone una sobrecarga adicional.

Sobrecarga de E/S de disco

El software de administración del rendimiento no supone una sobrecarga mayor del disco en el sistema. Normalmente, Glance no registra sus datos en el disco. El demonio del recopilador de Componente Performance Collection, `scope`, genera archivos del registro del disco, pero ARM 2.0 no afecta significativamente a su tamaño. El archivo de registro `logtran scope` se usa para almacenar datos de ARM.

Sobrecargade la CPU

Un programa instrumentado con llamadas ARM normalmente no se ejecutará más lento debido a las llamadas ARM. Esto supone que la cantidad de llamadas `arm_getid` es inferior a una llamada por segundo, y la cantidad de llamadas `arm_start` y `arm_stop` es inferior a unos pocos miles por segundo. Deben evitarse llamadas más frecuentes a la API de ARM.

La mayor parte de la sobrecarga adicional de la CPU para admitir ARM se produce dentro de los programas de herramientas de rendimiento y los mismos demonios. La sobrecarga de la CPU `midaemon` se eleva ligeramente, pero no más del 2% de lo que lo hacía con ARM 1.0. Si se ha solicitado que `midaemon` realice un seguimiento de las métricas de recursos por transacción, la sobrecarga por cada transacción puede ser dos veces más alta que si el seguimiento de las métricas de recursos por transacción estuviera desactivado. (Puede activar el seguimiento de las métricas de recurso por transacciones estableciendo la etiqueta `log transaction=resource` en el archivo `parm`). Además, la sobrecarga de la CPU en Glance y `scope` será un poco mayor en un sistema con aplicaciones instrumentadas con llamadas a ARM 2.0. Sólo las aplicaciones que están instrumentadas con llamadas a ARM 2.0 y usan mucho los correladores y/o métricas definidas por el usuario tendrán un importante impacto en el rendimiento en `midaemon`, `scope` o Glance.

Una condición de desbordamiento de `midaemon` puede producirse cuando el uso excede la memoria compartida predeterminada disponible. El resultado es:

- No hay códigos de retorno desde las llamadas ARM una vez que se produce la condición de desbordamiento.
- Se muestra una métrica incorrecta, incluidos nombres de procesos en blanco.
- Se registran errores en `status.mi` (por ejemplo, no hay espacio).

Sobrecargade la memoria

Los programas que están realizando llamadas a la API de ARM no tendrán un impacto importante en el tamaño del conjunto de la memoria virtual, excepto en el espacio usado para pasar el correlador de ARM 2.0 y la información de la métrica definida por el usuario. Estos búferes, que se explican en el manual *Application Response Measurement 2.0 API Guide* (Guía de la API de Application Response Measurement 2.0), no deben constituir una parte importante de los requisitos de memoria del proceso.

Hay una sobrecarga del tamaño del conjunto virtual en las herramientas de rendimiento para admitir el uso de ARM 2.0. El proceso `midaemon` crea un segmento de memoria compartida en el que los datos de ARM se mantienen internamente para que los usen Componente Performance Collection y GlancePlus. El tamaño de este segmento de memoria compartida ha crecido con respecto al tamaño de las versiones con ARM 1.0, para adaptarse a la posibilidad de que lo use ARM 2.0. De forma predeterminada, en la mayoría de los sistemas este segmento de memoria compartida tiene un tamaño aproximado de 11 megabytes. Este segmento no reside totalmente en la memoria física, a menos que sea necesario. Por tanto, no debería tener un impacto importante en la mayoría de los sistemas que no tengan antes problemas de memoria. La sobrecarga de la memoria de `midaemon` puede ajustarse mediante unos parámetros de inicio especiales (consulte la página del manual *midaemon*).

20 Introducción a las transacciones

En este capítulo se proporciona la información que necesita para empezar a realizar seguimientos de las transacciones y los objetivos de nivel de servicio. Para obtener información de referencia detallada, consulte el [Capítulo 19, Funcionamiento del seguimiento de transacciones](#). Consulte el [Capítulo 23, Ejemplos de seguimiento de transacciones](#) para ver ejemplos.

Antes de comenzar

Componente Performance Collection proporciona la biblioteca compartida `libarm.*` en las siguientes ubicaciones:

Plataforma	Ruta
IBM RS/6000	<code>/usr/lpp/perf/lib/</code>
Otras plataformas UNIX	<code>/opt/perf/lib/</code>

Si no tiene Componente Performance Collection instalado el sistema y `libarm.*` no existe en la ruta antes indicada para su plataforma, consulte [Ejemplos de opción de compilador C por plataforma](#) en la página 394, al final de este manual. Consulte también la sección sobre la biblioteca compartida de ARM (`libarm`) en *Application Response Measurement 2.0 API Guide* (Guía de la API de Application Response Measurement 2.0) para ver información acerca de cómo obtenerla. Para ver una descripción de `libarm`, consulte [Biblioteca ARM \(libarm\)](#) en la página 388, al final de este manual.

Configuración del seguimiento de transacciones

Siga este procedimiento para configurar el seguimiento de transacciones para su aplicación. Estos pasos se describen más detalladamente en el resto de esta sección.

- 1 Defina los SLO determinando las principales transacciones que desea monitorizar y el nivel de respuesta previsto (*opcional*).
- 2 Para monitorizar transacciones en Componente Performance Collection y Performance Manager, asegúrese de que el archivo Componente Performance Collection `parm` tiene el registro de transacciones activado. A continuación, inicie o reinicie Componente Performance Collection para leer el archivo `parm` actualizado.

La edición del archivo `parm` *no* es necesaria para ver las transacciones en GlancePlus. Sin embargo, `ttd` *debe* estar ejecutándose para ver las transacciones en GlancePlus. Al iniciar GlancePlus se iniciará automáticamente `ttd`.

- 3 Ejecute la aplicación que se haya instrumentado con llamadas a la API de ARM que se describen en este manual y en *Application Response Measurement 2.0 API Guide* (Guía de la API de Application Response Measurement 2.0).
- 4 Use Componente Performance Collection o Performance Manager para ver los datos de transacciones recopilados o usar GlancePlus para ver los datos actuales. Si los datos no están visibles en Performance Manager, cierre el origen de los datos y vuelva a conectarse a él.
- 5 Personalice el archivo de configuración, `ttd.conf`, para modificar la forma en que los datos de las transacciones de la aplicación se recopilan (*opcional*).
- 6 Después de realizar adiciones en el archivo `ttd.conf`, debe llevar a cabo los siguientes pasos para que las adiciones entren en vigor:
 - a Detenga todas las aplicaciones sobre plataformas ARM.
 - b Ejecute el comando `ttd -hup -mi` como **raíz**.Estas acciones vuelven a leer el archivo `ttd.conf` y registran las nuevas transacciones junto con los valores de `slo` y `range` con `ttd` y `midaemon`. La nueva lectura no cambiará los valores de `slo` ni `range` para ninguna transacción que estaba en el archivo `ttd.conf` antes de la nueva lectura.
- 7 Si necesita cambiar los valores de `slo` o `range` para transacciones existentes en el archivo `ttd.conf`, haga lo siguiente.
 - a Detenga todas las aplicaciones sobre plataformas ARM.
 - b Detenga el recopilador `scope` mediante `ovpa stop`.
 - c Detenga el uso de Glance.
 - d Detenga `ttd` con `ttd -k`.

Cuando haya realizado los cambios:

- a Reinicie `scope` con `ovpa start`.
- b Inicie las aplicaciones sobre plataformas ARM.

Definición de los objetivos de nivel de servicio

El primer paso para implementar el seguimiento de transacciones es determinar las principales transacciones necesarias para cubrir las expectativas de los clientes, así como el nivel de respuesta a las transacciones que es necesario. El nivel de respuesta necesario pasa a ser su objetivo de nivel de servicio (SLO). Defina el objetivo de nivel de servicio en el archivo de configuración, `ttd.conf`.

La definición de los objetivos de nivel de servicio puede ser tan simple como revisar el contrato de nivel de servicio (SLA) del departamento de informática para ver las transacciones necesarias para monitorizar el cumplimiento del SLA. Si no tiene ningún SLA, puede desear implementar uno. Sin embargo, no es necesario crear un SLA para realizar un seguimiento de las transacciones.

Modificación del archivo parm

Si es necesario, modifique el archivo Componente Performance Collection `parm` para agregar transacciones a la lista de elementos que se registrarán para usarlos con Performance Manager y Componente Performance Collection. Incluya la opción `transaction` en el parámetro `log` del archivo `parm`, como se muestra en el siguiente ejemplo:

```
log global application process transaction device=disk
```

El valor predeterminado para el parámetro `log transaction` es `no resource` y no `correlator`. Para activar la recopilación de datos de recursos (sólo HP-UX) o recopilación de datos del correlador, hay que especificar `log transaction=resource` o `log transaction=correlator`. Ambos se pueden registrar especificando `log transaction=resource, correlator`.

Antes de poder recopilar datos de transacciones para usarlos con Componente Performance Collection y Performance Manager, es necesario actualizar el archivo `parm` como se describe a continuación:

Estado de Componente Performance Collection	Comando para activar el seguimiento de transacciones
En ejecución	<code>ovpa restart</code>
No ejecutándose	<code>ovpa start</code>

Recopilación de datos de transacciones

Inicie la aplicación. El demonio del seguimiento de transacciones, `ttd`, y el demonio de la interfaz de medición, `midaemon`, recopilan y sincronizan los datos de transacciones para su aplicación a medida que se ejecuta. Los datos se almacenan en el segmento de la memoria compartida de `midaemon`, donde Componente Performance Collection o GlancePlus pueden usarlo. Consulte [Monitorización de datos de rendimiento](#) en la página 371 para obtener información acerca del uso de cada una de estas herramientas para ver datos de transacciones para su aplicación.

Control de errores

Debido a consideraciones de rendimiento, no todas las llamadas a la API de ARM o Transaction Tracker problemáticas devuelven errores en tiempo real. Algunos ejemplos de cuándo no se devuelven errores como se preveía son:

- llamar a `arm_start` con un parámetro `id` incorrecto, como una variable no inicializada
- llamar a `arm_stop` sin una llamada `arm_start` previa correcta

Componente Performance Collection: para depurar estas situaciones cuando se instrumentan aplicaciones con llamadas a ARM, ejecute la aplicación durante el tiempo suficiente para generar y recopilar bastante cantidad de datos de transacciones. Recopile estos datos con Componente Performance Collection y, a continuación, use el comando `export` del programa `extract` para exportar los datos desde el archivo `logtran`. Examine los datos para ver si todas las transacciones se registran como está previsto. Además, compruebe el archivo `/var/opt/perf/status.ttd` para ver si hay posibles errores.

GlancePlus: para depurar estas situaciones cuando se instrumentan aplicaciones con llamadas a ARM, ejecute la aplicación durante el tiempo suficiente para generar bastante cantidad de datos de transacciones; a continuación, use GlancePlus para ver si todas las transacciones aparecen como estaba previsto.

Límites en transacciones exclusivas

En función de sus requisitos específicos del sistema y la configuración del núcleo, puede haber un límite en el número de transacciones exclusivas permitidas en su aplicación. Este límite suele ser varios miles de llamadas `arm_getid` exclusivas.

El número de transacciones exclusivas puede exceder del límite cuando el segmento de memoria compartida usado por `midaemon` está lleno. Si esto ocurre, aparece un mensaje de desbordamiento en GlancePlus. Aunque no aparece ningún mensaje en Componente Performance Collection, no se registrarán los datos de las transacciones nuevas posteriores. (Sin embargo, compruebe `/var/opt/perf/status.scope` para ver si hay un mensaje de desbordamiento). Los datos de las nuevas transacciones posteriores no estarán visibles en GlancePlus. Las transacciones que ya se han registrado continuarán registrándose e incluyéndose en informes. La métrica `GBL_TT_OVERFLOW_COUNT` en GlancePlus informa del número de nuevas transacciones que no pudieron medirse.

Esta situación puede solucionarse deteniendo e iniciando el proceso `midaemon` con la opción `-smdvss` para especificar un tamaño de segmento de memoria compartida mayor. El tamaño actual del segmento de memoria compartida puede comprobarse mediante el comando `midaemon -sizes`. Para obtener más información acerca de cómo optimizar `midaemon` en su sistema, consulte la página del manual sobre *midaemon*.

Personalización del archivo de configuración (opcional)

Después de ver los datos de transacciones de su aplicación, puede desear personalizar el archivo de configuración de transacciones, `/var/opt/perf/ttd.conf`, para modificar la forma en que se recopilan los datos de transacciones para la aplicación. Esto es opcional, porque el archivo de configuración predeterminado, `ttd.conf`, funcionará con todas las transacciones definidas en la aplicación. Si decide personalizar el archivo `ttd.conf`, complete esta tarea en los mismos sistemas en los que ejecuta la aplicación. Debe haber iniciado una sesión como **raíz** para modificar `ttd.conf`.

Consulte el [Capítulo 19, Funcionamiento del seguimiento de transacciones](#) para obtener información acerca de las palabras clave del archivo de configuración: `tran`, `range` y `slo`. A continuación se proporcionan algunos ejemplos de cómo se usa cada palabra clave:

```
tran=Example:   tran=answerid
                tran=answerid*
                tran=*

range=Example:  range=2.5,4.2,5.0,10.009

slo=Example:    slo=4.2
```

Personalice su archivo de configuración para que incluya todas las transacciones y los atributos asociados. Tenga en cuenta que el uso de la palabra clave `range` o `slo` debe estar precedida por la palabra clave `tran`. A continuación se muestra un ejemplo del archivo `ttd.conf`.

```
tran=*
tran=my_first_transaction slo=5.5

[answerid]
tran=answerid1 range=2.5, 4.2, 5.0, 10.009 slo=4.2

[orderid]
tran=orderid1 range=1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0
```

Si necesita realizar adiciones al archivo `ttd.conf`:

- Detenga todas las aplicaciones sobre plataformas ARM.
- Ejecute el comando `ttd -hup -mi` como **raíz**.

Las anteriores acciones vuelven a leer el archivo `ttd.conf` y registran las nuevas transacciones junto con los valores de `slo` y `range` con `ttd` y `midaemon`. La nueva lectura no cambiará los valores de `slo` ni `range` para ninguna transacción que estaba en el archivo `ttd.conf` antes de la nueva lectura.

Si necesita cambiar los valores de `slo` o `range` para transacciones existentes en el archivo `ttd.conf`, haga lo siguiente.

- 1 Detenga todas las aplicaciones sobre plataformas ARM.
- 2 Detenga el compilador `scope` mediante **`ovpa stop`**.
- 3 Detenga el uso de Glance.
- 4 Detenga `ttd` con **`ttd -k`**.

Cuando haya realizado los cambios:

- 1 Reinicie `scope` con **`ovpa start`**.
- 2 Inicie las aplicaciones sobre plataformas ARM.

Monitorización de datos de rendimiento

Puede usar los siguientes productos de administración del rendimiento y recursos para monitorizar los datos de transacciones: Componente Performance Collection, Performance Manager y GlancePlus.

... con Componente Performance Collection

Al recopilar y registrar datos durante largos períodos, Componente Performance Collection permite analizar el rendimiento del sistema a lo largo del tiempo y realizar análisis de tendencias detallados. Los datos de Componente Performance Collection pueden verse con Performance Manager Agent o exportarse para usarlos con otras herramientas de monitorización del rendimiento, contabilidad, modelado y planificación.

Con el programa `extract` de Componente Performance Collection, los datos pueden exportarse para usarlos con hojas de cálculo y programas de análisis. Los datos también puede extraerse para archivarlos y analizarlos.

Componente Performance Collection y ttd deben estar ejecutándose para monitorizar los datos de transacciones en Componente Performance Collection. Iniciar Componente Performance Collection con la secuencia de comandos ovpa garantiza que los procesos ttd y midaemon necesarios para ver los datos de transacciones en GlancePlus se inician en el orden correcto.

... con Performance Manager

Performance Manager importa los datos de Componente Performance Collection y proporciona la posibilidad de convertir esos datos a un formato gráfico o numérico personalizado. Mediante Performance Manager, puede realizar análisis de tendencias históricas de los datos de transacciones para conseguir previsiones más precisas.

Puede seleccionar **TRANSACTION** en la ventana Class List para un origen de datos en Performance Manager y, a continuación, graph transaction metrics para varias transacciones. Para obtener más información, consulte la ayuda en línea de Performance Manager, a la que puede acceder mediante el menú Help de Performance Manager. Si no ve las transacciones que esperaba en Performance Manager, cierre el origen de los datos actual y vuelva a conectarse a él.

... con GlancePlus

Monitorizar sistemas con GlancePlus ayuda a identificar los cuellos de botella de los recursos y proporciona información del rendimiento inmediata acerca del sistema informático. GlancePlus tiene una ventana de seguimiento de transacciones en la que se muestra información acerca de todas las transacciones que ha definido y una ventana de gráficos de transacciones en la que se muestra información específica acerca de una sola transacción. Por ejemplo, puede ver cómo está funcionando cada transacción con respecto al SLO que ha definido. Para obtener más información acerca de cómo usar GlancePlus, consulte la ayuda en línea, a la que puede acceder mediante el menú Help.

Alarmas

Puede generar alarmas basadas en los datos de transacciones con los siguientes productos de administración del rendimiento y recursos: Componente Performance Collection, Performance Manager y GlancePlus.

... con Componente Performance Collection

Para generar alarmas con Componente Performance Collection, debe definir las condiciones de las alarmas en su archivo de definiciones de alarmas: `alarmdef`. Puede configurar Componente Performance Collection para que le notifique una situación de alarma de diversas formas; por ejemplo, enviando un mensaje de correo electrónico o realizando una llamada a su buscapersonas.

Para realizar una comprobación de la sintaxis del archivo `alarmdef`, debe tener datos registrados para este nombre de aplicación y nombre de transacción en los archivos del registro, o bien tener los nombres registrados en el archivo `ttd.conf`.

Hay una limitación al definir una condición de alarma basada en una transacción que tenga un guión (-) en su nombre. Para que esta limitación no le afecte, use el comando `ALIAS` en el archivo `alarmdef` para cambiar el nombre de la transacción.

... con GlancePlus

Puede configurar la sintaxis del asesor para que genere alarmas basadas en el rendimiento de las transacciones. Por ejemplo, cuando se cumple una condición de alarma, puede indicar a GlancePlus que muestre información a `stdout`, que ejecute un comando de UNIX (como `mailx`) o que cambie el botón de alarma de la ventana principal de GlancePlus a amarillo o rojo. Para obtener más información acerca de las alarmas en GlancePlus, elija **On This Window** en el menú Help de la ventana Edit Adviser Syntax.

21 Mensajes de seguimiento de transacciones

Se devuelven los códigos de error de la [Tabla 2](#) y los puede usar el desarrollador de aplicaciones al instrumentar una aplicación con llamadas a la API de Application Response Measurement (ARM) o de Transaction Tracker:

Tabla 2 Códigos de error

Código de error	Valor n° error	Significado
-1	EINVAL	Argumentos no válidos
-2	EPIPE	ttd (demonio de registro) no se está ejecutando
-3	ESRCH	El nombre de la transacción no se encontró en el archivo ttd.conf
-4	EOPNOTSUPP	Versión del sistema operativo no admitida

Cuando se está ejecutando una aplicación instrumentada con llamadas a la API de ARM o Transaction Tracker, los códigos de retorno de los errores que se hayan producido procederán probablemente del demonio de seguimiento de transacciones, ttd. El demonio de la interfaz de medida, midaemon, no produce ningún código de retorno de error.

Si se produce un error de midaemon, consulte el archivo `/var/opt/perf/status.mi` para obtener más información.

22 Métricas de transacciones

El agente de ARM incluido como componente compartido de GlancePlus y de Componente Performance Collection produce distintas métricas de transacciones. Consulte lo siguiente para obtener una lista completa de las métricas y sus descripciones:

- Para obtener información sobre las métricas GlancePlus instaladas, utilice la ayuda en línea de GlancePlus o consulte *GlancePlus for HP-UX Dictionary of Performance Metrics* en las ubicaciones siguientes:

En UNIX/Linux en `<InstallDir>/paperdocs/gp/C/` como `gp-metrics.txt`.

`InstallDir` es el directorio en el que se ha instalado Componente Performance Collection.

- Para obtener información sobre las métricas de Componente Performance Collection instaladas en plataformas específicas, consulte los archivos *HP Operations Agent Dictionary of Operating System Performance Metrics* de la plataforma en las ubicaciones siguientes:

En UNIX/Linux en `<InstallDir>/paperdocs/ovpa/C/` como `met<plataforma>.txt`.

En Windows en `%ovinstalldir%paperdocs\ovpa\C` como `met<plataforma>.txt`.

23 Ejemplos de seguimiento de transacciones

Este capítulo contiene un ejemplo de pseudocódigo sobre cómo se puede instrumentar una aplicación con llamadas a la API de ARM, para que las transacciones definidas en la aplicación se puedan monitorizar con Componente Performance Collection o GlancePlus. Este ejemplo de pseudocódigo se corresponde con el escenario de procesamiento de pedidos en tiempo real descrito en [Capítulo 18, Descripción del seguimiento de transacciones](#)

En este capítulo se incluyen varios archivos de configuración de transacciones de ejemplo, incluyendo el que se corresponde con el escenario de procesamiento en pedidos en tiempo real.

Pseudocódigo para el procesamiento de pedidos en tiempo real

Este ejemplo de pseudocódigo incluye las llamadas a la API de ARM utilizadas para definir transacciones para el escenario de procesamiento de pedidos en tiempo real descrito en [Capítulo 18, Descripción del seguimiento de transacciones](#). Se debe procesar esta rutina *cada vez* que un operador conteste el teléfono para gestionar un pedido del cliente. En este ejemplo, las líneas que contienen las llamadas a la API de ARM están resaltadas en **negrita**.

```
routine answer calls()
{
*****
* Register the transactions if first time in          *
*****
  if (transactions not registered)
  {
    appl_id = arm_init("Order Processing Application", "*", 0,0,0)
    answer_phone_id = arm_getid(appl_id,"answer_phone","1st tran",0,0,0)
    if (answer_phone_id < 0)
      REGISTER OF ANSWER_PHONE FAILED - TAKE APPROPRIATE ACTION
    order_id = arm_getid(appl_id,"order","2nd tran",0,0,0)
    if (order_id < 0)
      REGISTER OF ORDER FAILED - TAKE APPROPRIATE ACTION
    check_id = arm_getid(appl_id,"check_db","3rd tran",0,0,0)
    if (check_id < 0)
      REGISTER OF CHECK DB FAILED - TAKE APPROPRIATE ACTION
    update_id = arm_getid(appl_id,"update","4th tran",0,0,0)
    if (update_id < 0)
      REGISTER OF UPDATE FAILED - TAKE APPROPRIATE ACTION

  } if transactions not registered
*****
* Main transaction processing loop
*****
  while (answering calls)
  {
```

```

        if (answer_phone_handle = arm_start(answer_phone_id,0,0,0) < -1)
            TRANSACTION START FOR ANSWER_PHONE NOT REGISTERED
*****
* At this point the answer_phone transaction has      *
* started. If the customer does not want to order, *
* end the call; otherwise, proceed with order.      *
*****
        if (don't want to order)
            arm_stop(answer_phone_handle,ARM_FAILED,0,0,0)
        GOOD-BYE - call complete
        else
            {
*****
* They want to place an order - start an order now *
*****
                if (order_handle = arm_start(order_id,0,0,0) < -1)
                    TRANSACTION START FOR ORDER FAILED
                take order information: name, address, item, etc.
*****
* Order is complete - end the order transaction      *
*****
                    if (arm_stop(order_handle,ARM_GOOD,0,0,0) < -1)
                        TRANSACTION END FOR ORDER FAILED
*****
* order taken - query database for availability      *
*****
                        if (query_handle = arm_start(query_id,0,0,0) < -1)
                            TRANSACTION QUERY DB FOR ORDER NOT REGISTERED
                        query the database for availability
*****
* database query complete - end query transaction    *
*****
                            if (arm_stop(query_handle,ARM_GOOD,0,0,0) < -1)
                                TRANSACTION END FOR QUERY DB FAILED
*****
* If the item is in stock, process order, and        *
* update inventory.                                  *
*****
                            if (item in stock)
                                if (update_handle = arm_start(update_id,0,0,0) < -1)
                                    TRANSACTION START FOR UPDATE NOT REGISTERED
                                update stock
*****
* update complete - end the update transaction      *
*****
                                    if (arm_stop(update_handle,ARM_GOOD,0,0,0) < -1)
                                        TRANSACTION END FOR ORDER FAILED
*****
* Order complete - end the call transaction          *
*****
                                        if (arm_stop(answer_phone_handle,ARM_GOOD,0,0,0) < -1)
                                            TRANSACTION END FOR ANSWER_PHONE FAILED

```

```

    } placing the order
GOOD-BYE - call complete

sleep("waiting for next phone call...zzz...")
} while answering calls
arm_end(appl_id, 0,0,0)
} routine answer calls

```

Ejemplos de archivos de configuración

En esta sección se incluyen algunos ejemplos del archivo de configuración de transacciones, `/var/opt/perf/ttd.conf`. Para obtener más información sobre el archivo `ttd.conf` y las palabras clave del archivo de configuración, consulte [Capítulo 19, Funcionamiento del seguimiento de transacciones](#)

Ejemplo 1 (ejemplo de pseudocódigo de procesamiento de pedidos)

```

# The "*" entry below is used as the default if none of the
# entries match a registered transaction name.

tran=* range=0.5,1,1.5,2,3,4,5,6,7 slo=1
tran=answer_phone* range=0.5,1,1.5,2,3,4,5,6,7 slo=5
tran=order* range=0.5,1,1.5,2,3,4,5,6,7 slo=5
tran=query_db* range=0.5,1,1.5,2,3,4,5,6,7 slo=5

```

Ejemplo 2

```

# The "*" entry below is used as the default if none of the
# entries match a registered transaction name.

tran=* range=1,2,3,4,5,6,7,8 slo=5

# The entry below is for the only transaction being
# tracked in this application. The "*" has been inserted
# at the end of the tran name to catch any possible numbered
# transactions. For example "First_Transaction1",
# "First_Transaction2", etc.

tran=First_Transaction* range=1,2.2,3.3,4.0,5.5,10 slo=5.5

```

Ejemplo 3

```

# The "*" entry below is used as the default if none of the
# entries match a registered transaction name.

tran=*
tran=Transaction_One range=1,10,20,30,40,50,60 slo=30

```

Ejemplo 4

```
tran=FactoryStor* range=0.05, 0.10, 0.15 slo=3

# The entries below shows the use of an application name.
# Transactions are grouped under the application name. This
# example also shows the use of less than 10 ranges and
# optional use of "slo."

[Inventory]
tran=In_Stock range=0.001, 0.004, 0.008
tran=Out_Stock range=0.001, 0.005
tran>Returns range=0.1, 0.3, 0.7

[Pers]
tran=Acctg range=0.5, 0.10, slo=5
tran=Time_Cards range=0.010, 0.020
```

24 Funciones avanzadas

En este capítulo se describe cómo Componente Performance Collection utiliza las siguientes funciones de la API de ARM 2.0:

- tipos de datos
- métricas definidas por el usuario
- instrumentación de `scope`

Cómo se usan los tipos de datos

En la [Tabla 3](#) se describe cómo se usan los tipos de datos en Componente Performance Collection. Es un complemento de "Data Type Definitions" (Definiciones de tipos de datos) en la sección "Temas avanzados" de la guía *Application Response Measurement 2.0 API Guide* (Guía de la API de Application Response Measurement 2.0).

Tabla 3 Uso de los tipos de datos de Componente Performance Collection

ARM_Counter32	Los datos se registran como un entero de 32 bits.
ARM_Counter64	Los datos se registran como un entero de 32 bits con conversión de tipos.
ARM_CntrDivr32	Realiza el cálculo y registra el resultado como un entero de 32 bits.
ARM_Gauge32	Los datos se registran como un entero de 32 bits.
ARM_Gauge64	Los datos se registran como un entero de 32 bits con conversión de tipos.
ARM_GaugeDivr32	Realiza el cálculo y registra el resultado como un entero de 32 bits.
ARM_NumericID32	Los datos se registran como un entero de 32 bits.
ARM_NumericID64	Los datos se registran como un entero de 32 bits con conversión de tipos.
ARM_String8	Ignorado.
ARM_String32	Ignorado.

Componente Performance Collection no registra datos de cadena. Como Componente Performance Collection registra los datos cada cinco minutos, y lo que se registra es el resumen de la actividad para ese intervalo, no puede resumir las cadenas proporcionadas por la aplicación.

Componente Performance Collection registra el mínimo, máximo y promedio de las seis primeras métricas definidas por el usuario. Si la aplicación instrumentada para ARM pasa un Counter32, un String8, un NumericID 32, un Gauge32, un Gauge64, un Counter64, un NumericID64, un String32 y un GaugeDivr32, Componente Performance Collection registra los valores Min, Max y Average en el intervalo de cinco minutos para Counter32, NumericID32, Gauge32, Gauge64, NumericID32 y NumericID64 como números enteros de 32 bits. String8 y String32 se ignoran porque las cadenas no se pueden resumir en Componente Performance Collection. GaugeDivr32 también se ignora porque sólo se registran las seis primeras métricas definidas por el usuario que se pueden utilizar. (Para más ejemplos, consulte la siguiente sección, [Métricas definidas por el usuario](#).)

Métricas definidas por el usuario

Esta sección es un complemento de "Application-Defined Metrics" (Métricas definidas por la aplicación) en la sección "Temas avanzados" de la guía *Application Response Measurement 2.0 API Guide (Guía de la API de Application Response Measurement 2.0)*. Contiene algunos ejemplos sobre cómo Componente Performance Collection controla las métricas definidas por el usuario (referidas como métricas definidas por la aplicación en ARM). En los ejemplos de la [Tabla 4](#) se muestra lo que se registra si el programa pasa los tipos de datos siguientes.

Tabla 4 Ejemplos de lo que se registra con tipos de datos específicos del programa

...lo que pasa el programa	...lo que se registra
EJEMPLO 1 String8 Counter32 Gauge32 CntrDivr32	Counter32 Gauge32 CntrDivr32
EJEMPLO 2 String32 NumericID32 NumericID64	NumericID32 NumericID64

Tabla 4 Ejemplos de lo que se registra con tipos de datos específicos del programa

...lo que pasa el programa	...lo que se registra
EJEMPLO 3 NumericID32 String8 NumericID64 Gauge32 String32 Gauge64	NumericID32 NumericID64 Gauge32 Gauge64
EJEMPLO 4 String8 String32	(nada)
EJEMPLO 5 Counter32 Counter64 CntrDivr32 Gauge32 Gauge64 NumericID32 NumericID64	Counter32 Counter64 CntrDivr32 Gauge32 Gauge64 NumericID32

Como Componente Performance Collection no puede resumir cadenas, éstas no se registran.

En el ejemplo 1, sólo se registran los valores counter, gauge y counter divisor.

En el ejemplo 2, sólo se registran los valores numeric.

En el ejemplo 3, sólo se registran los valores numeric y gauge.

En el ejemplo 4, no se registra nada.

En el ejemplo 5, como sólo se registran las seis primeras métricas definidas por el usuario, NumericID64 no se registra.

Instrumentación de scope

El recopilador de datos `scope` se ha instrumentado con llamadas a la API de ARM. Cuando se inicia `Componente Performance Collection`, `scope` comienza automáticamente a registrar dos transacciones, `Scope_Get_Process_Metrics` y `Scope_Get_Global_Metrics`. Ambas transacciones se encontrarán en la aplicación `HP Performance Tools`.

Los datos de transacción se registran cada cinco minutos, por tanto se encontrará que se han completado cinco transacciones `Get Process` (una transacción por minuto) durante cada intervalo. La transacción `Scope_Get_Process_Metrics` se instrumenta en torno al procesamiento de los datos de proceso. Si hay 200 procesos en el sistema, la transacción `Scope_Get_Process_Metrics` tardará más tiempo que con 30 procesos.

La transacción `Scope_Get_Global_Metrics` está instrumentada en torno a la recopilación de todos los datos de cinco minutos, incluyendo los datos globales, como `global`, `application`, `disk`, `transaction` y otros tipos de datos.

Para detener el registro de los datos de transacciones de proceso y globales, hay que suprimir o convertir en comentarios las entradas para las transacciones `scope` en el archivo `ttd.conf`.

25 Bibliotecas de transacciones

Este apéndice incluye los componentes siguientes:

- La biblioteca Application Response Measurement (`libarm`)
- Ejemplos de la opción de compilador C por plataforma
- La biblioteca Application Response Measurement NOP (`libarmNOP`)
- Uso de contenedores Java

Biblioteca ARM (libarm)

Con Componente Performance Collection y GlancePlus, el entorno es configurado de manera que resulte fácil la operación de compilación y el uso de la utilidad ARM facility.

Las bibliotecas necesarias para el desarrollo se encuentran en `/opt/perf/lib/`. Consulte la sección siguiente de este apéndice para obtener información específica sobre la compilación.

Las bibliotecas enunciadas en la [Tabla 5](#) existen en instalaciones HP-UX 11.11 y fuera de Componente Performance Collection e instalación GlancePlus:

Tabla 5 HP-UX 11.11 y fuera de Componente Performance Collection y archivos de biblioteca GlancePlus

<code>/opt/perf/lib/</code>	<code>libarm.0</code>	Biblioteca compartida compatible para ARM HP-UX 10.X (insegura para subprocesos). Si se ejecuta un programa en HP-UX 11 que ha sido vinculado en 10.20 con <code>-larm</code> , el cargador 11.0 referenciará automáticamente esta biblioteca.
	<code>libarm.1</code>	Biblioteca compartida compatible HP-UX 11 (segura para subprocesos). Esta biblioteca será referenciada por programas que hayan sido vinculados con <code>-larm</code> o versiones HP-UX. Si un programa vinculado con 10.20 referencia esta biblioteca, (por ejemplo, si no ha sido vinculado con <code>-L /opt/perf/lib</code> , puede que aborte con un error del tipo <code>"/usr/lib/dld.sl: Unresolved symbol: _thread_once (code) from libtt.sl"</code> .
	<code>libarm.sl</code>	Vínculo simbólico a <code>libarm.1</code>
	<code>libarmNOP.sl</code>	Biblioteca compartida “sin-operaciones” para ARM (las llamadas API ocurren pero no hacen nada; usada para pruebas y en sistemas que no tienen instalado Componente Performance Collection.
<code>/opt/perf/examples/arm</code>	<code>libarmjava.sl</code>	Biblioteca compartida de 32 bits para ARM.
<code>/opt/perf/examples/arm/arm64</code>	<code>libarmjava.sl</code>	Biblioteca compartida de 64 bits para ARM.

Tabla 5 HP-UX 11.11 y fuera de Componente Performance Collection y archivos de biblioteca GlancePlus

/opt/perf/ lib/pa20_64/	Nota: estos archivos serán referenciados automáticamente por programas compilados HP-UX 11 con la opción de compilador +DD64.	
libarm.sl	Biblioteca compartida de 64 bits para ARM.	
libarmNOP.sl	Biblioteca compartida “sin-operaciones” de 64 bits para ARM (las llamadas API ocurren pero no hacen nada; usada para pruebas y en sistemas que no tienen instalado Componente Performance Collection).	

Los archivos de biblioteca adicionales enunciados en la [Tabla 6](#) existen en instalaciones IA64 HP-UX:

Tabla 6 Archivos de biblioteca HP-UX IA64

/opt/perf/lib/hpux32/	libarm.so.1	Biblioteca compartida IA64 de 32 bits para ARM.
/opt/perf/lib/hpux64/	libarm.so.1	Biblioteca compartida IA64 de 64 bits para ARM.
/opt/perf/examples/arm	libarmjava. so	Biblioteca compartida de 32 bits para ARM.
/opt/perf/examples/ arm/arm64	libarmjava. so	Biblioteca compartida de 64 bits para ARM.

Dado que la biblioteca ARM hace llamadas a HP-UX que pueden cambiar de una versión de sistema operativo a otra, los programas deberían vincularse con la versión de biblioteca compartida mediante `-larm`. No se admite compilar una aplicación que ha sido instrumentada con llamadas API ARM y vincular con la versión archivada de la librería ARM (archivo `-Wl,-a`). Para obtener mayor información, consulte [Demonio de seguimiento de transacciones \(ttd\)](#) en la página 357 en el capítulo 2.

Los archivos de biblioteca existentes en sistemas operativos AIX con Componente Performance Collection y la instalación GlancePlus son los siguientes:

Tabla 7 Archivos de biblioteca AIX

/usr/lpp/perf/lib/	libarm.a	Biblioteca ARM compartida de 32 bits (segura para subprocesos). Esta biblioteca es referenciada por programas vinculados con -larm.
/usr/lpp/perf/lib	libarmNOP.a	Biblioteca compartida de 32 bits para ARM. Esta biblioteca es usada para pruebas en sistemas que no tienen instalado Performance Agent/ Componente Performance Collection.
/usr/lpp/perf/lib64/	libarm.a	Biblioteca ARM compartida de 64 bits (segura para subprocesos). Esta biblioteca es referenciada por programas vinculados con -larm.
/usr/lpp/perf/lib64	libarmNOP.a	Biblioteca compartida de 32 bits para ARM. Esta biblioteca es usada para pruebas en sistemas que no tienen instalado Performance Agent/ Componente Performance Collection.
/usr/lpp/perf/examples/arm	libarmjava.a	Biblioteca compartida de 32 bits para ARM.
/usr/lpp/perf/examples/arm/arm64	libarmjava.a	Biblioteca compartida de 64 bits para ARM.
/usr/lpp/perf/lib/	libarmns.a	Biblioteca ARM archivada de 32 bits. Equivalente en funcionalidad a libarm.a. de 32 bits.
/usr/lpp/perf/lib64/	libarmns.a	Biblioteca ARM archivada de 64 bits. Equivalente en funcionalidad a libarm.a. de 64 bits.

Los archivos de biblioteca existentes en sistemas operativos Solaris con Componente Performance Collection y la instalación GlancePlus son los siguientes:

Tabla 8 Archivos de biblioteca Solaris para programas de 32 bits

/opt/perf/lib/	libarm.so	Biblioteca ARM compartida de 32 bits (segura para subprocesos). Esta biblioteca es referenciada por programas vinculados con -larm.
	libarmNOP.so	Biblioteca compartida de 32 bits para ARM. Esta biblioteca es usada para pruebas en sistemas que no tienen instalado Componente Performance Collection.

Tabla 9 Archivos de biblioteca Solaris para programas de 64 bits Sparc

/opt/perf/lib/ sparc_64/	libarm.so	Biblioteca ARM compartida de 64 bits (segura para subprocesos). Esta biblioteca es referenciada por programas vinculados con -larm.
	libarmNOP.so	Biblioteca compartida de 64 bits para ARM. Esta biblioteca es usada para pruebas en sistemas que no tienen instalado Performance Agent/Componente Performance Collection.
/opt/perf/ examples/arm	libarmjava.so	Biblioteca compartida de 32 bits para ARM.
/opt/perf/ examples/arm/ arm64	libarmjava.so	Biblioteca compartida de 64 bits para ARM.

Tabla 10 Archivos de biblioteca Solaris para programas de 64 bits x86

/opt/perf/lib/ x86_64/	libarm.so	Biblioteca ARM compartida de 64 bits (segura para subprocesos). Esta biblioteca es referenciada por programas vinculados con -larm.
	libarmNOP.so	Biblioteca compartida de 64 bits para ARM. Esta biblioteca es usada para pruebas en sistemas que no tienen instalado Performance Agent.
/opt/perf/ examples/arm	libarmjava.so	Biblioteca compartida de 32 bits para ARM.
/opt/perf/ examples/arm/ arm64	libarmjava.so	Biblioteca compartida de 64 bits para ARM.



Se deberán compilar programas de 64 bits con el parámetro de línea de comandos `-xarch=generic64` junto con los otros parámetros proporcionados para programas de 32 bits.

Los archivos de biblioteca existentes en sistemas operativos Linux con Componente Performance Collection e instalación GlancePlus son los siguientes:

Tabla 11 Archivos de biblioteca Linux

/opt/perf/lib/	libarm.so	Biblioteca ARM compartida de 32 bits (segura para subprocesos). Esta biblioteca es referenciada por programas vinculados con -larm.
	libarmNOP.so	Biblioteca compartida de 32 bits para ARM. Esta biblioteca es usada para pruebas en sistemas que no tienen instalado Componente Performance Collection.

Tabla 11 Archivos de biblioteca Linux

/opt/perf/lib64/	libarm.so	Biblioteca ARM compartida de 64 bits (segura para subprocesos). Esta biblioteca es referenciada por programas vinculados con -larm.
	libarmNOP.so	Biblioteca compartida de 64 bits para ARM. Esta biblioteca es usada para pruebas en sistemas que no tienen instalado Componente Performance Collection.
/opt/perf/examples/arm	libarmjava.so	Biblioteca compartida de 32 bits para ARM.
/opt/perf/examples/arm/arm64	libarmjava.so	Biblioteca compartida de 64 bits para ARM.



Para Linux 2.6 IA 64 bits 32 bits, libarm.so y libarmjava.so no han sido implementados.

Ejemplos de opción de compilador C por plataforma

El archivo de inclusión `arm.h` se encuentra en `/opt/perf/include/`. Para comodidad del usuario, este archivo también se encuentra accesible a través de un vínculo simbólico en `/usr/include/`. Esto significa que el usuario no necesitará usar `“-I/opt/perf/include/”` (aunque puede usarlo si lo desea). Asimismo, `libarm` reside en `/opt/perf/lib/` pero ha sido vinculado en `/usr/lib/`. Al crear aplicaciones sobre plataformas ARM deberá usarse siempre `“-L/opt/perf/lib/”`.

- Para Linux:

El ejemplo siguiente muestra un comando de compilación para un programa C usando el API ARM.

```
cc myfile.c -o myfile -I /opt/perf/include -L
-Xlinker -rpath -Xlinker /opt/perf/lib
```

- Para programas de 64 bits en Linux:

```
cc -m64 myfile.c -o myfile -I /opt/perf/include -L -Xlinker -rpath
-Xlinker /opt/perf/lib64
```

- Para HP-UX:

En el caso de versiones HP-UX 11.2x en plataformas IA64, cambie el parámetro `-L` por `-L/opt/perf/lib a-L/opt/perf/lib/hpux32` para compilaciones de programas de 32 bits IA ARM, y por `-L/opt/perf/lib/hpux64` para programas de 64 bits IA.

El ejemplo siguiente muestra un comando de compilación para un programa C usando el API ARM.

```
cc myfile.c -o myfile -I /opt/perf/include -L /opt/perf/lib -larm
```

- Para Sun Solaris:

El siguiente ejemplo se aplica a Componente Performance Collection y GlancePlus en Sun Solaris:

```
cc myfile.c -o myfile -I /opt/perf/include -L /opt/perf/lib -larm -lnsl
```

- Para programas Sparc de 64 bits en Sun Solaris:

El siguiente ejemplo se aplica a Componente Performance Collection y programas de 64 bits en Sun Solaris:

```
cc -xarch=generic64 myfile.c -o myfile -I /opt/perf/include -L /opt/perf/
lib/sparc_64 -larm -lnsl
```

- Para programas de 64 bits x86 en Sun Solaris:

El siguiente ejemplo se aplica a Performance agent y programas de 64 bits en Sun Solaris:

```
cc -xarch=generic64 myfile.c -o myfile -I /opt/perf/include -L /opt/perf/
lib/x86_64 -larm -lnsl
```

- Para AIX de IBM:

La ubicación de archivo en AIX de IBM difiere de otras plataformas (se usa `/usr/lpp/perf/` en lugar de `/opt/perf/`), por lo que el ejemplo para AIX de IBM es diferente de los ejemplos de otras plataformas:

```
cc myfile.c -o myfile -I /usr/lpp/perf/include -L /usr/lpp/perf/lib -larm
```

- Para programas de 64 bits en AIX de IBM:

El siguiente ejemplo se aplica a Performance agent y programas de 64 bits en AIX de IBM:

```
cc -q64 myfile.c -o myfile -I /usr/lpp/perf/include -L /usr/lpp/perf/lib64
-larm
```



Para compiladores C++, el indicador `-D_PROTOTYPES` puede necesitar ser agregado al comando de compilación con objeto de referenciar las declaraciones apropiadas en el archivo `arm.h`.

Biblioteca ARM NOP

La biblioteca “sin operaciones (llamada `libarmNOP.*` en la que `*` es `sl`, `so`, o `a`, dependiendo de la plataforma OS) viene con Componente Performance Collection y Glance incorporados. Esta biblioteca compartida no hace nada más que comunicar estados válidos para cada llamada de API ARM. Ello permite que una aplicación que ha sido instrumentada con ARM se ejecute en un sistema en el que se encuentre instalado Componente Performance Collection o GlancePlus.

Para ejecutar la aplicación instrumentada ARM en un sistema en el que no ha sido instalado Componente Performance Collection o GlancePlus, copie la biblioteca NOP y llámela `libarm.sl` (`libarm.so` o `libarm.a` dependiendo de la plataforma) en el directorio apropiado (normalmente, `/<InstallDir>/lib/`). Si Componente Performance Collection o GlancePlus ha sido instalado, sobrescribirá esta biblioteca NOP sustituyéndola por la biblioteca correcta (la cual no es eliminada mientras que los otros archivos sí). Ello asegurará que no se aborten los programas instrumentados al eliminar Componente Performance Collection o GlancePlus del sistema.

Uso de los contenedores Java

Los contenedores de la interfaz Java Native (JNI) son funciones creadas para la comodidad del usuario que permiten a las aplicaciones Java llamar a la API ARM2.0 de HP. Estos contenedores (`armapi.jar`) están incluidos en los programas de muestra ARM ubicados en el directorio `/<InstallDir>/examples/arm/`. `InstallDir` es el directorio en el que se ha instalado Componente Performance Collection.

Ejemplos

Existen ejemplos de contenedores Java en el directorio `/<InstallDir>/examples/arm/`. Esta ubicación también contiene un archivo README que explica la función de cada contenedor.

Configurar una aplicación (arm_init)

Para configurar una nueva aplicación es necesario crear una nueva instancia de `ARMApplication` y pasar el nombre y la descripción por esa API. Cada aplicación deberá ser identificada con un nombre único. La clase `ARMApplication` usa la función C – `arm_init`.

Sintaxis:

```
ARMApplication myApplication =  
new ARMApplication("name","description");
```

Configurar una transacción (arm_getid)

Para configurar una transacción el usuario podrá decidir si desea o no usar métrica definida por el usuario (UDMs). El contenedor Java usa la función C – `arm_getid`.

Configurar una transacción con UDM

Si se desea usar UDM, deberá definirse primero una nueva `ARMTranDescription`. `ARMTranDescription` crea el búfer de datos de `arm_getid`. (Consulte también el ejemplo `jprimeudm.java`.)

Sintaxis:

```
ARMTranDescription myDescription =  
new ARMTranDescription("transactionName","details");
```

Si no desea usar detalles, use otro constructor:

Sintaxis:

```
ARMTranDescription myDescription =  
new ARMTranDescription("transactionName");
```

Agregar la métrica

Métrica 1-6:

Sintaxis:

```
myDescription.addMetric(metricPosition, metricType,  
metricDescription);
```

Parámetros:

`metricPosition`: 1-6

`metricType`: `ARMConstants.ARM_Counter32`
`ARMConstants.ARM_Counter64` `ARMConstants.ARM_CntrDivr32`
`ARMConstants.ARM_Gauge32` `ARMConstants.ARM_Gauge64`
`ARMConstants.ARM_GaugeDivr32` `ARMConstants.ARM_NumericID32`
`ARMConstants.ARM_NumericID64` `ARMConstants.ARM_String8`

Métrica 7:

Sintaxis:

```
myDescription.addStringMetric("description");
```

Then you can create the Transaction:

Sintaxis:

```
myApplication.createTransaction(myDescription);
```

Configurar los datos de métrica

Métrica 1-6:

Sintaxis:

```
myTransaction.setMetricData(metricPosition, metric);
```

Ejemplos de "Métrica"

```
ARMGauge32Metric metric = new ARMGauge32Metric(start);  
ARMCounter32Metric metric = new ARMCounter32Metric(start);  
ARMCntrDivr32Metric metric = new ARMCntrDivr32Metric(start, 1000);
```

Métrica 7:

Sintaxis:

```
myTransaction.setStringMetricData(text);
```

Configurar una transacción sin UDM

Si se configura una transacción sin UDM, la nueva transacción podrá crearse inmediatamente. El usuario podrá decidir si desea o no especificar detalles.

Con detalles

Sintaxis:

```
ARMTransaction myTransaction =  
myApplication.createTransaction("Transactionname", "details");
```

Sin detalles

Sintaxis:

```
ARMTransaction myTransaction =  
myApplication.createTransaction("Transactionname");
```

Configurar una instancia de transacción

Para configurar una instancia de transacción, cree una nueva instancia de `ARMTransactionInstance` con el método `createTransactionInstance()` de `ARMTransaction`.

Sintaxis:

```
ARMTransactionInstance myTranInstance =  
myTransaction.createTransactionInstance();
```

Iniciar una instancia de transacción (`arm_start`)

Para iniciar una instancia de transacción el usuario podrá decidir si desea o no usar correladores. Los siguientes métodos llaman la función `C - arm_start` con los parámetros permitentes.

Iniciar la instancia de transacción con correladores

Al usar correladores el usuario deberá distinguir entre recibir y enviar un correlador.

Solicitar un correlador

Si la instancia de transacción del usuario quiere solicitar un correlador, la llamada será como sigue (consulte también el ejemplo `jcorrelators.java`).

Sintaxis:

```
int status = myTranInstance.startTranWithCorrelator();
```

Pasar el correlador primario

Si el usuario ya dispone de correlador de una transacción anterior y desea enviar el correlador a su transacción, la sintaxis será como sigue:

Sintaxis

```
int status = startTran(parent);
```

Parámetro

`parent` es el correlador enviado. En la transacción anterior el correlador de instancia de transacción se puede obtener con el método `getCorrelator()`.

Solicitar y pasar el correlador primario

Si el usuario ya dispone de correlador de una transacción anterior y desea enviarlo a su transacción y solicitar un correlador, la sintaxis será como sigue:

Sintaxis:

```
int status = myTranInstance.startTranWithCorrelator(parent);
```

Parámetro:

parent es el correlador enviado. En la transacción anterior el correlador de instancia de transacción se puede obtener con el método `getCorrelator()`.

Recuperar la información del correlador

El usuario podrá recuperar el correlador de instancia de transacción usando el método `getCorrelator()` como sigue a continuación:

Sintaxis:

```
ARMTranCorrelator parent = myTranInstance.getCorrelator();
```

Iniciar la instancia de transacción sin correladores

Si no se usan correladores, el usuario podrá iniciar la instancia de transacción como sigue a continuación:

Sintaxis:

```
int status = myTranInstance.startTran();
```

`startTran` devuelve un controlador único al estado, el cual es usado para la actualización y parada.

Actualizar los datos de instancia de transacción

El usuario podrá actualizar los UDM de su instancia de transacción las veces que lo desee entre el inicio y la parada. Esta parte de los contenedores llama la función `C - arm_update` con los parámetros pertinentes.

Actualizar los datos de instancia de transacción con UDM

Cuando se actualizan los datos de la instancia de transacción con UDM, se deberán en primer lugar establecer los nuevos datos de métrica. Por ejemplo:

```
metric.setData(value) for ARM_Counter32 ARM_Counter64, ARM_Gauge32,  
ARM_Gauge64, ARM_NumericID32, ARM_NumericID64  
metric.setData(value,value) for ARM_CntrDivr32 and , ARM_GaugeDivr32  
metric.setData(string) for ARM_String8 and ARM_String32
```

A continuación el usuario podrá establecer a nuevos los datos de métrica (como en los ejemplos de la sección [Configurar los datos de métrica](#)) y llamar la actualización:

Sintaxis:

```
myTranInstance.updateTranInstance();
```

Actualizar los datos de instancia de transacción sin UDM

Cuando se actualizan los datos de la instancia de transacción sin UDM, se acaba de llamar la actualización. Ello envía un “latido” que indica que la instancia de transacción aún se está ejecutando.

Sintaxis:

```
myTranInstance.updateTranInstance();
```

Proporcionar un búfer privado de aplicación opaco y grande

Si se desea usar el segundo formato de búfer, deberá pasarse la matriz de bytes al método de actualización. (Consulte la Guía API *Application Response Measurement 2.0*).

Sintaxis:

```
myTranInstance.updateTranInstance(byteArray);
```

Detener la instancia de transacción (arm_stop)

Para detener la instancia de transacción el usuario podrá decidir si desea o no detenerla con o sin actualización de métrica.

Detener la instancia de transacción con actualización de métrica

Para detener la instancia de transacción con actualización de métrica, llame el método `stopTranInstanceWithMetricUpdate`.

Sintaxis:

```
myTranInstance.stopTranInstanceWithMetricUpdate  
(transactionCompletionCode);
```


Parámetro:

El código de finalización de transacción podrá ser:

<code>ARMConstants.ARM_GOOD.</code>	Use este valor cuando la operación transcurra con normalidad y según lo esperado.
<code>ARMConstants.ARM_ABORT.</code>	Use este valor cuando se produzca un fallo grave en el sistema.
<code>ARMConstants.ARM_FAILED.</code>	Use este valor en aplicaciones en las que la transacción transcurra con normalidad, pero no se generen resultados.

Estos métodos usan la función C – `arm_stop` con los parámetros solicitados.

Detener la instancia de transacción sin actualización de métrica

Para detener la instancia de transacción sin actualización de métrica, use el método `stopTranInstance`.

Sintaxis:

```
myTranInstance.stopTranInstance(transactionCompletionCode);
```

Usar la transacción completada

Los contenedores Java pueden usar la llamada `arm_complete_transaction`. Esta `InstallDir>/examples/llamada` puede utilizarse para marcar el final de una transacción que ha durado un número determinado de nanosegundos. Ello permite la integración en tiempo real de respuestas de transacción medidas fuera del agente ARM.

Además de marcar el final de una instancia de transacción, puede proporcionarse información adicional sobre la transacción (UDM) en el búfer de datos opcional.

(Consulte también el ejemplo `jcomplete.java`).

Usar transacción completada con UDM:

Sintaxis:

```
myTranInstance.completeTranWithUserData(status, responseTime);
```

Parámetros:

status	<ul style="list-style-type: none">• ARMConstants.ARM_GOOD. Use este valor cuando la operación transcurra con normalidad y según lo esperado.• ARMConstants.ARM_ABORT Use este valor cuando se produzca un fallo grave en el sistema.• ARMConstants.ARM_FAILED Use este valor en aplicaciones en las que la transacción transcurra con normalidad, pero no se generen resultados.
responseTime	Éste es el tiempo de respuesta de la transacción en nanosegundos.

Usar transacción completada sin UDM:

Sintaxis:

```
myTranInstance.completeTran(status, responseTime);
```

Documentación adicional

Para obtener información adicional sobre las clases Java, consulte la carpeta doc en el directorio `<InstallDir>/examples/arm/`, la cual incluye documentación html para cada clase Java. Comience con `index.htm`.

26 Registro y rastreo

Puede diagnosticar y solucionar problemas en HP Operations Agent mediante los mecanismos de registro y rastreo. HP Operations Agent almacena mensajes de error, advertencias y generales en archivos de registro para analizarlos fácilmente.

El mecanismo de rastreo ayuda a rastrear problemas concretos en el funcionamiento del agente; puede transferir los archivos de rastreo generados mediante el mecanismo de rastreo al soporte técnico de HP para su análisis.

Registro

HP Operations Agent escribe mensajes de advertencia y error, así como notificaciones informativas en el archivo `System.txt`, en el nodo. El contenido del archivo `System.txt` revela que la gente está funcionando como está previsto. Puede encontrar el archivo `System.txt` en la siguiente ubicación:

En Windows

`%ovdatadir%\log`

En UNIX/Linux

`/var/opt/OV/log`

Además, HP Operations Agent agrega detalles de estado de Componente Performance Collection y coda a los siguientes archivos:

En Windows

- `%ovdatadir%\status.scope`
- `%ovdatadir%\status.perfalarm`
- `%ovdatadir%\status.ttd`
- `%ovdatadir%\status.mi`
- `%ovdatadir%\status.perfd-<puerto>`



En este caso, `<puerto>` es el puerto usado por `perfd`. De forma predeterminada, `perfd` usa el puerto 5227. Para cambiar el puerto predeterminado de `perfd`, consulte [Configuración del componente RTMA](#) en la página 51.

- `%ovdatadir%\log\coda.txt`

En UNIX/Linux

- /var/opt/perf/status.scope
- /var/opt/perf/status.perfalarm
- /var/opt/perf/status.ttd
- /var/opt/perf/status.mi
- /var/opt/perf/status.perfd
- *Sólo en vMA.* /var/opt/perf/status.viserver
- /var/opt/OV/log/coda.txt

Configuración de la directiva de registro

El archivo `System.txt` puede crecer hasta un tamaño de 1 MB; después, el agente empieza a registrar los mensajes en una nueva versión del archivo `System.txt`. Puede configurar la directiva de registro de mensajes de HP Operations Agent para restringir el tamaño del archivo `System.txt`.

Para modificar la directiva de registro predeterminada, siga estos pasos:

- 1 Inicie una sesión en el nodo.
- 2 Vaya a la ubicación siguiente:

En Windows

`%ovdatadir%conf\xpl\log`

En UNIX/Linux

`/var/opt/OV/conf/xpl/log`

- 3 Abra el archivo `log.cfg` con un editor de texto.
- 4 Los parámetros `BinSizeLimit` y `TextSizeLimit` controlan el tamaño en bytes y el número de caracteres del archivo `System.txt`. De forma predeterminada, los dos parámetros se establecen en 1000000 (1 MB y 1000000 caracteres). Cambie los valores predeterminados a los valores deseados.
- 5 Guarde el archivo.
- 6 Reinicie Componente de monitorización de operaciones con los siguientes comandos:
 - a `ovc -kill`
 - b `ovc -start`

Rastreo

Antes de iniciar el rastreo de una aplicación de HP Operations Agent, debe realizar una serie de tareas previas, incluida la identificación de la aplicación correcta que se va a rastrear, la configuración del tipo de rastreo y la generación de un archivo de configuración de rastreo (si es necesario).

Antes de iniciar el rastreo de un proceso de HP Operations Agent, realice las siguientes tareas:

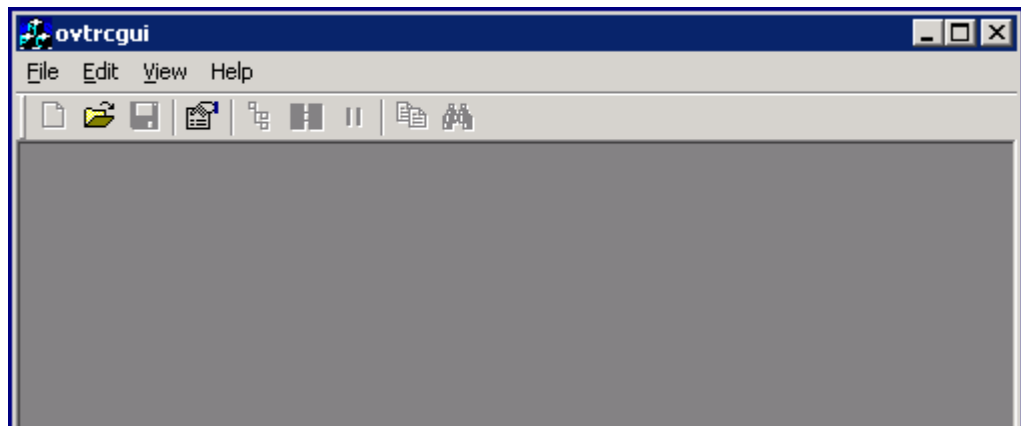
- 1 [Identificación de la aplicación](#) en la página 405
- 2 [Establecimiento del tipo de rastreo](#) en la página 407
- 3 *Opcional.* [Creación del archivo de configuración](#) en la página 410

Identificación de la aplicación

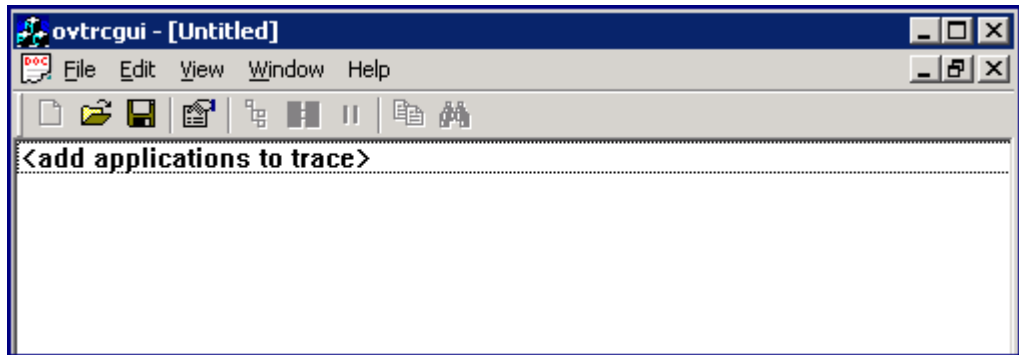
En el sistema administrado, identifique las aplicaciones de HP Software que desee rastrear. Use la opción `ovtrccfg -vc` para ver los nombres de todas las aplicaciones con el rastreo activado y los componentes y categorías definidos para cada una de ellas.

Si lo prefiere, puede usar la utilidad `ovtrcgui` para ver la lista de aplicaciones con el rastreo activado. Para usar la utilidad `ovtrcgui` para ver la lista de aplicaciones con el rastreo activado, siga estos pasos.

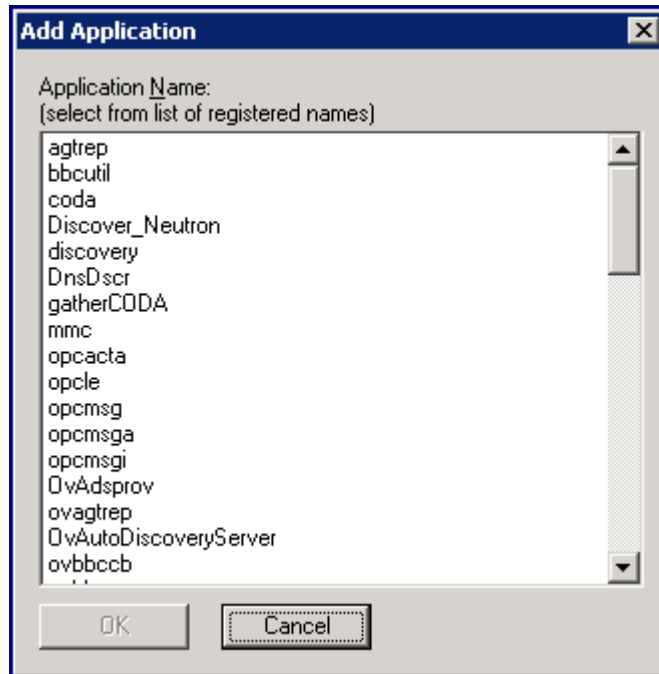
- 1 Ejecute el archivo `ovtrcgui.exe` en el directorio `%OvInstallDir%\support`. Se abre la ventana `ovtrcgui`.



- 2 En la ventana ovtrcgui, haga clic en **File** → **New** → **Trace Configuration**. Se abre un nuevo editor de configuración de rastreo.



- 3 En la ventana ovtrcgui, haga clic en **Edit** → **Add Application**. Si lo prefiere, haga clic con el botón derecho en el editor y, a continuación, haga clic en **Add Application**. Se abre la ventana Add Application.



La ventana Add Application presenta una lista de las aplicaciones con rastreo activado que están disponibles.

Establecimiento del tipo de rastreo

Antes de activar el mecanismo de rastreo, decida y establezca el tipo de rastreo que desea configurar con una aplicación. Para establecer el tipo de rastreo, siga estos pasos:

Determine el tipo de rastreo (estático o dinámico) que desea configurar y siga estos pasos:

- 1 Vaya a la ubicación `<data_dir>/conf/xpl/trc/`
- 2 Localice el archivo `<nombre_aplicación>.ini`. Si el archivo está presente, vaya al paso [paso 3](#). Si el archivo `<nombre_aplicación>.ini` no está presente, siga estos pasos:
 - Cree un archivo con un editor de texto.
 - Agregue las siguientes propiedades al archivo en el orden indicado: DoTrace, UpdateTemplate, and DynamicTracing.
 - ▶ No enumere las propiedades en una sola línea. Enumere cada propiedad en una nueva línea. Por ejemplo:

```
DoTrace=  
UpDateTemplate=  
DynamicTracing=
```
 - Guarde el archivo.
- 3 Abra el archivo `<nombre_aplicación>.ini` con un editor de texto.
- 4 Para activar el **rastreo estático**, asegúrese de que la propiedad DoTrace se establece en ON y la propiedad DynamicTracing se establece en OFF.
- 5 Para activar el **rastreo dinámico**, asegúrese de que las propiedades DoTrace y DynamicTracing se establecen en ON.
- 6 Asegúrese de que la propiedad UpdateTemplate esta establecida en ON.
- 7 Guarde el archivo.

En el caso de la configuración del rastreo dinámico, puede habilitar el mecanismo de rastreo incluso después de que se inicie la aplicación. En el caso de la configuración del rastreo estático, debe habilitar el mecanismo de rastreo antes de que se inicie la aplicación.

Introducción al archivo de configuración de rastreo

Sintaxis

```
TCF Version <número_versión>  
APP: "<nombre_aplicación>"  
SINK: File "<nombre_archivo>" "maxfiles=[1..100];maxsize=[0..1000];"  
TRACE: "<nombre_componente>" "<nombre_categoria>" <lista_palabras_clave>
```

Cada línea de la sintaxis se describe detalladamente en las siguientes secciones.

TCF Version

La línea TCF especifica que se trata de un archivo de configuración del rastreo y el número de versión del archivo. Distingue entre y debe especificarse exactamente como se muestra a continuación:

Sintaxis:

```
TCF Version <número_versión>
```

Ejemplo:

```
TCF Version 1.1
```

APP

En la línea de la aplicación se define el nombre de la aplicación que se va a rastrear. Debe empezar con `APP`, seguido por un signo de dos puntos (`:`) y un espacio; el nombre de la aplicación debe estar entre comillas ("`...`"). Se pueden especificar varias aplicaciones para el rastreo. Repita este patrón para cada aplicación que desee rastrear.

Sintaxis:

```
APP: "<nombre_aplicación>"
```

Ejemplo:

```
APP: "dbmanager"
```

```
APP: "opcmsg"
```

```
APP: "poller"
```

SINK

En la línea del receptor se especifica el archivo de destino al que se dirigirá el resultado del rastreo. El destino debe ser un archivo en el mismo equipo. La línea debe empezar por `SINK:` `FILE`. Los argumentos en la línea deben estar separados por espacios.

La línea `SINK: FILE` tiene dos argumentos.

El primer argumento es el nombre del archivo de destino, que debe estar entre comillas ("`...`").

El segundo argumento son las opciones del tipo de receptor adicional. Las opciones deben estar entre comillas ("`...`"), y cada opción debe estar seguida por un signo de punto y coma (`;`).

Las opciones para `File` de tipo de receptor son:

- `maxfiles=n`
- `maxsize=n`



En un archivo de configuración de rastreo existente, puede observar la opción `force`. Esta opción no se admite con esta versión de la utilidad de rastreo y no tiene ningún efecto en el mecanismo de rastreo. Puede ignorar esta opción al configurar el mecanismo de rastreo.

maxfiles

La opción `maxfiles` va seguida de un valor entero entre 1 y 100. Esta opción permite especificar el número de archivos de registro de rastreo históricos que se mantendrá. Cada vez que se inicia una aplicación para rastrear el archivo, se realiza una copia de seguridad del archivo anterior (si existe) agregando `".001"` al nombre y cambiando el nombre del archivo. Si ya había un archivo `".001"`, se cambia el nombre a `".002"` y así sucesivamente. El mismo esquema de copia de seguridad está en vigor si el archivo de registro actual alcanza el tamaño máximo.

maxsize

La opción `maxsize` va seguida de un valor entero o de coma flotante entre 0 y 1000 que especifica el espacio en disco máximo en megabytes (MB) que se usará para cada archivo.

Si el último bloque del resultado de rastreo escrita en el archivo hace que el archivo tenga un tamaño mayor que el máximo especificado, el siguiente resultado generará una copia de seguridad, cerrará el archivo de salida actual y creará un archivo de resultados. Un valor de 0 es un caso especial que permite que el archivo crezca hasta quedarse sin espacio en disco.



Use la herramienta `ovtrcmon` o `ovtrcgui` para ver el contenido del archivo de destino al que se ha dirigido el resultado del rastreo. El formato no está ordenado si el archivo se abre con un editor de texto estándar.

Sintaxis:

```
SINK: File "<nombre_archivo>" "maxfiles=[1..100];maxsize=[0..1000];"
```

Ejemplo:

```
SINK: File "C:\\TEMP\\Output.trc" "maxfiles=10;maxsize=100;"
```

TRACE

La línea de rastreo debe empezar por `TRACE`, seguido por un signo de dos puntos (`:`) y un espacio (). Los argumentos en la línea deben estar separados por espacios.

El primer argumento es el nombre del componente de rastreo, que debe estar entre comillas ("`...`"). Pueden especificarse varios componentes.

El segundo argumento es el nombre de la categoría de rastreo; también debe estar entre comillas ("`...`"). Pueden especificarse varios componentes. Si está usando una de las categorías estándar del código, se asigna al valor de cadena que especifique aquí. Para la asignación exacta de las constantes de categoría estándar a los valores de las cadenas, consulte la documentación correspondiente al lenguaje (C++, Java).

Sintaxis:

```
TRACE: "<nombre_componente>" "<nombre_categoria>" <lista_palabras_clave>
```

Ejemplo:

```
TRACE: "database" "Parms" Error Info Warn  
TRACE: "xpl.io" "Trace" Info
```

Puede usar "*" como nombre del componente, como nombre de la categoría o como ambos. Es útil al usar aplicaciones en el modo en el que leen su información de configuración directamente de un archivo.

Cuando una aplicación intenta determinar la configuración de `component A` y `category B`, mira primero si la configuración contiene una definición de rastreo explícita para este par. Si la definición de rastreo está ahí, usa esta configuración. De lo contrario, busca si hay una configuración para `component A` y `category *`. Si la hay, usa esta configuración. De lo contrario, busca si hay una configuración para `component * and category *`. Si la hay, usa esta configuración. De lo contrario, no se activa el rastreo.

Los demás parámetros son una lista de variables de las opciones de palabras clave. Al menos una de las palabras, Error, Info o Warn, debe estar en la lista. Las palabras clave admitidas son:

Tabla 12 Palabras clave de rastreo

Palabra clave	Descripción
Error	Habilitar rastreos marcados como errores.
Warn	Habilitar rastreos marcados como advertencias.
Info	Habilitar rastreos marcados como información.
Soporte técnico	Habilitar el rastreo normal. El resultado del rastreo incluye mensajes informativos, advertencias y mensajes de error. Se recomienda esta opción para problemas que surjan durante la solución de problemas. El rastreo se puede habilitar para un largo periodo ya que la sobrecarga para capturar el resultado del rastreo es mínima con esta opción.

Ejemplo de archivos de configuración del rastreo

```
TCF Version 3.2
APP: "dbmanager"
SINK: File "C:\\TEMP\\Output.trc" "maxfiles=10;maxsize=100;"
TRACE: "DbManager" "Parms" Error Info Warn Developer
```

Creación del archivo de configuración

Si desea habilitar el mecanismo de rastreo sin la ayuda de un archivo de configuración, sáltese esta sección y vaya a [Habilitación del rastreo y visualización de mensajes de rastreo con las herramientas de la línea de comandos](#) en la página 416.

Puede crear el archivo de configuración del rastreo con la herramienta de la línea de comandos `ovtrccfg`, con un editor de texto o con la utilidad `ovtrcgui` (sólo en los nodos de Windows).

Uso de la herramienta de la línea de comandos

Ejecute el siguiente comando para generar un archivo de configuración del rastreo:

```
ovtrccfg -app <nombre_aplicación> [-cm <nombre_componente>] [-sink <nombre_archivo>] -gc <nombre_archivo_configuración>
```

El comando crea el archivo de configuración con los detalles de las aplicaciones y componentes que desea rastrear.

Uso de un editor de texto

Si desea crear manualmente el archivo de configuración con un editor de texto, siga estos pasos:

- 1 Con un editor de texto, cree un archivo.
- 2 Especifique el número de versión del archivo de configuración al principio con el siguiente formato:

```
TCF Version <número_versión>
```

Por ejemplo:

```
TCF Version 1.0
```

- 3 Especifique la aplicación que desea rastrear en el formato siguiente:

```
APP <nombre_aplicación>
```

Por ejemplo:

```
APP "coda"
```

- 4 Especifique la publicación de destino en la que se almacenarán los archivos de rastreo en el formato siguiente:

```
SINK: File "<nombre_archivo>"  
"maxfiles=<máx_archivos>;maxsize=<tamaño_máx>"
```

- ▶ El nombre del archivo de rastreo debe tener la extensión `trc`. Asegúrese de especificar la ruta completa del archivo de resultados del rastreo en el formato siguiente:

```
<unidad>:\\<directorio>\\<nombre_archivo>
```

Por ejemplo:

```
SINK: File "C:\\TEMP\\Output.trc" "maxfiles=10;maxsize=100;"
```

Si no especifica ningún valor para el parámetro `SINK`, el mecanismo de rastreo empieza a colocar los archivos de resultado de rastreos en el directorio principal de la aplicación rastreada.

- 5 Especifique el componente y la categoría que desea rastrear en el formato siguiente:

```
TRACE: "<nombre_componente>" "<nombre_categoría>" "<lista_palabras_clave>"
```

- ▶ Si desea rastrear varios componentes o categorías, agregue varias instrucciones `TRACE` con saltos de línea.

Por ejemplo:

```
TRACE: "bbc.cb" "Parms" Error Info Warn
```

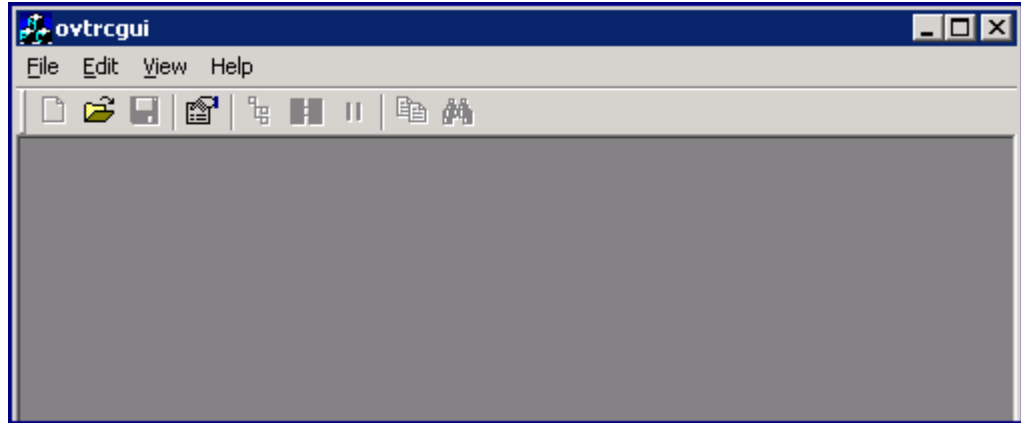
```
TRACE: "bbc.https.server" "Trace" Info
```

- 6 Guarde el archivo con la extensión `tcf`.

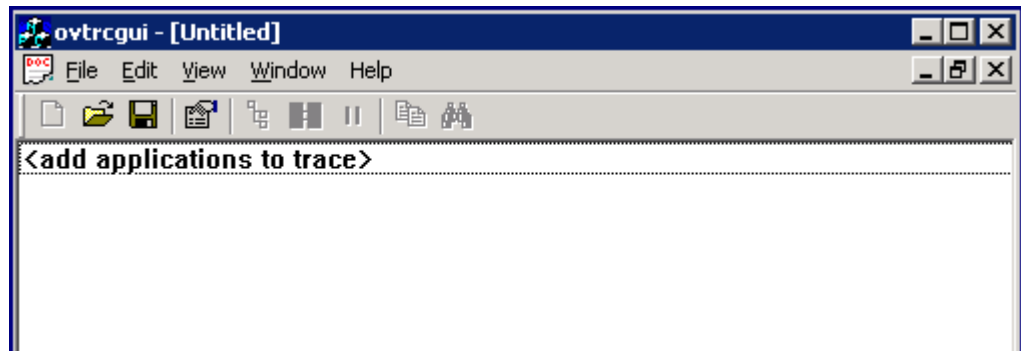
Uso de la GUI de rastreo

En los nodos de Windows, puede usar la GUI de rastreo (la utilidad `ovtrcgui`) para crear el archivo de configuración del rastreo. Para usar esta utilidad y crear un archivo de configuración del rastreo, siga estos pasos:

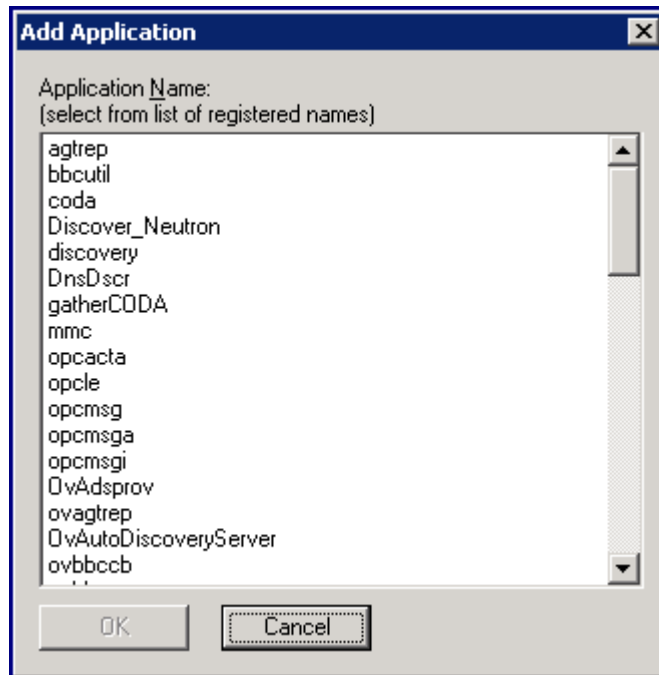
- 1 Ejecute el archivo `ovtrcgui.exe` en el directorio `%OvInstallDir%\support`. Se abre la ventana `ovtrcgui`.



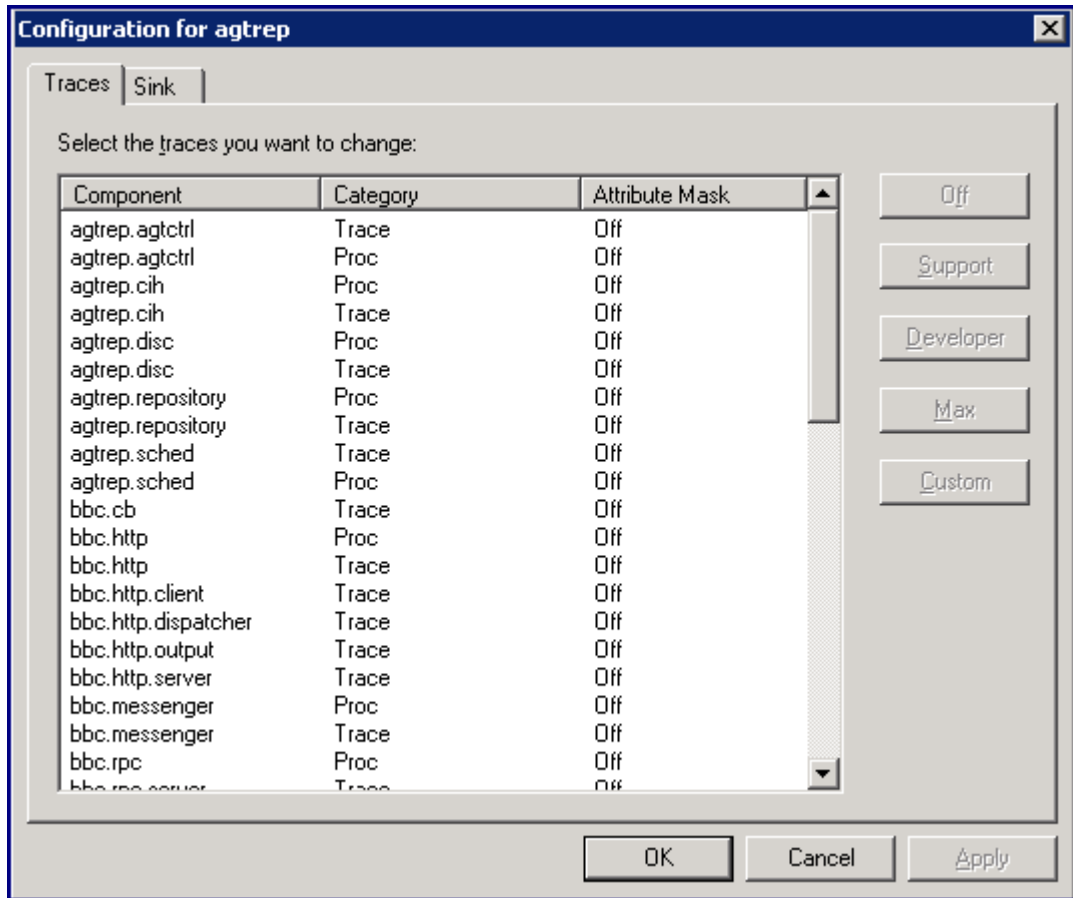
- 2 En la ventana `ovtrcgui`, haga clic en **File** → **New** → **Trace Configuration**. Se abre un nuevo editor de configuración de rastreo.



- 3 En la ventana ovtrgui, haga clic en **Edit** → **Add Application**. Si lo prefiere, haga clic con el botón derecho en el editor y, a continuación, haga clic en **Add Application**. Se abre la ventana Add Application.



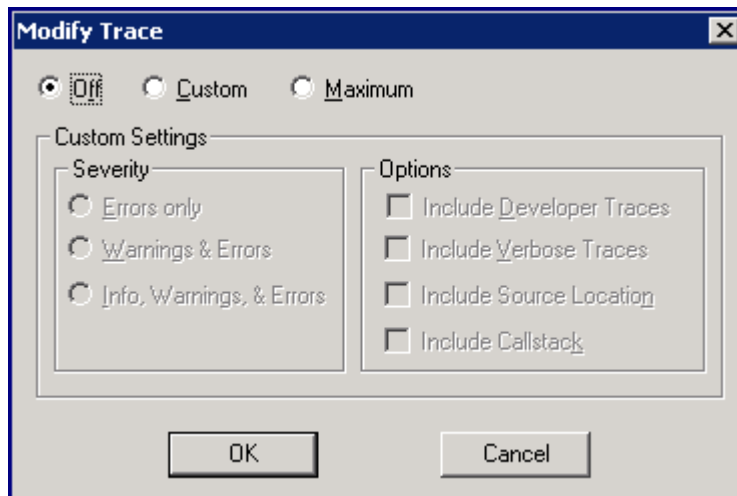
- 4 Seleccione la aplicación que desea rastrear y haga clic en **OK**. Se abre la ventana Configuration for <aplicación>.



En la pestaña Traces de la ventana Configuration for <aplicación> se enumeran todos los componentes y categorías de la aplicación seleccionada. De forma predeterminada, el rastreo de todos los componentes y categorías está establecido en Off.

- 5 En la pestaña Traces, haga clic en un par de componente categoría y, a continuación, en uno de los siguientes botones:
 - **Support:** Haga clic aquí para reunir los mensajes de rastreo marcados como notificaciones informativas.
 - **Developer:** Haga clic aquí para reunir los mensajes de rastreo marcados como notificaciones informativas, junto con todos los rastreos del desarrollador.
 - **Max:** Haga clic aquí para establecer el nivel máximo de rastreo.

- **Custom:** Al hacer clic en Custom, aparece la ventana Modify Trace.



En la ventana Modify Trace, seleccione las opciones personalizadas, los niveles de rastreo y las opciones que desee y haga clic en **OK**.

- ▶ En la ventana Configuration for <aplicación>, puede hacer clic en **Off** para desactivar el rastreo para un par componente-categoría.

6 Haga clic en **OK**.

7 Vaya a la pestaña Sink.

8 Especifique el nombre del archivo de resultados de rastreo en el cuadro de texto File Name. La extensión del archivo debe ser `.trc`.

- ▶ Especifique la ruta completa del archivo `.trc`.

9 Especifique el número de archivos históricos en la lista desplegable (consulte [maxfiles](#) en la página 408).

10 Especifique el tamaño de archivo máximo en la lista desplegable (consulte [maxsize](#) en la página 409).

11 Haga clic en **Apply**.

12 Haga clic en **OK**.

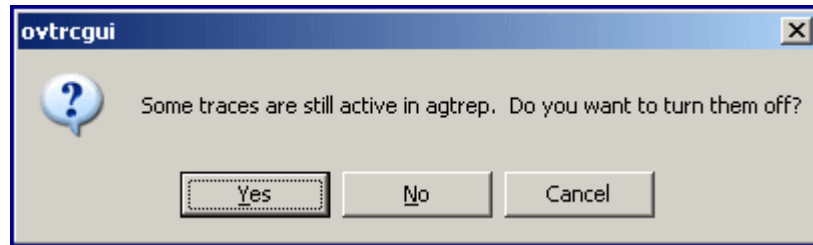
- ▶ La utilidad `ovtrcgui` habilita el mecanismo de rastreo al hacer clic en **OK**.

13 Haga clic en **File** → **Save**. Se abre el cuadro de diálogo Save As.

14 En el cuadro de diálogo Save As, vaya a una ubicación adecuada, especifique el nombre del archivo de configuración del rastreo con la extensión `.tcf` en el cuadro de texto del nombre del archivo y haga clic en **Save**.

La utilidad `ovtrcgui` guarda el nuevo archivo de configuración del rastreo en la ubicación especificada, con el nombre especificado, y habilita el mecanismo de rastreo de acuerdo con la configuración especificada en el archivo. Puede abrir el archivo de configuración del rastreo con la utilidad `ovtrcgui` y agregar nueva información de configuración.

- 15 Si intenta cerrar el editor de configuración del rastreo o la ventana `ovtrcgui`, aparece el siguiente mensaje:



- 16 Si hace clic en **No**, el mecanismo del rastreo continúa rastreando las aplicaciones configuradas en el sistema. Si hace clic en **Yes**, la utilidad `ovtrcgui` deshabilita inmediatamente el mecanismo de rastreo.

Habilitación del rastreo y visualización de mensajes de rastreo con las herramientas de la línea de comandos

El procedimiento indicado a continuación cubre la secuencia general de pasos necesaria para habilitar el rastreo. Para habilitar el mecanismo de rastreo, siga estos pasos:

- 1 Realice una solicitud de configuración del rastreo mediante `ovtrccfg`.

```
ovtrccfg -cf <nombre_archivo_configuración>
```

donde <nombre_archivo_configuración> es el nombre del archivo de configuración del rastreo creado en [Creación del archivo de configuración](#) en la página 410.

- Si no desea usar un archivo de configuración del rastreo, puede habilitar el rastreo con el siguiente comando:

```
ovtrccfg -app <aplicación> [-cm <componente>]
```

- 2 Si configura el mecanismo de rastreo estático, inicie la aplicación que desea rastrear.
- 3 Ejecute los comandos específicos de la aplicación necesarios para duplicar el problema que desea rastrear. Cuando se ha duplicado el comportamiento deseado, se puede detener el rastreo.
- 4 Realice una solicitud de monitorización del rastreo mediante `ovtrcmon`.

Para monitorizar los mensajes de rastreo, ejecute uno de los siguientes comandos o un comando similar con opciones del comando `ovtrcmon` adicionales:

- Para monitorizar los mensajes de rastreo de `/opt/OV/bin/trace1.trc` y dirigirlos a un archivo con formato de texto:

```
ovtrcmon -fromfile /opt/OV/bin/trace1.trc -tofile /tmp/  
traceout.txt
```

- Para ver los mensajes de rastreo de `/opt/OV/bin/trace1.trc` con formato detallado:

```
ovtrcmon -fromfile /opt/OV/bin/trace1.trc -verbose
```


- Para ver los mensajes de rastreo de `/opt/OV/bin/trace1.trc` con formato detallado y dirigir el mensaje de rastreo a un archivo:

```
ovtrcmon -fromfile /opt/OV/bin/trace1.trc -short > /tmp/traces.trc
```

- 5 Para detener o deshabilitar el rastreo mediante `ovtrccfg`, ejecute el siguiente comando:
ovtrccfg -off
- 6 Recopile el archivo de configuración de rastreo y los archivos de resultados de rastreos. Evalúe los mensajes de rastreo o empaquete los archivos para transferirlos al soporte técnico online de HP Software para su evaluación. Puede haber varias versiones de los archivos de resultados de rastreo en el sistema. La opción `Maxfiles` permite que el mecanismo de rastreo genere varios archivos de resultados de rastreo. Estos archivos tienen la extensión `.trc` y el sufijo `n` (donde `n` es un número entero entre 1 y 99999).

Habilitación del rastreo y visualización de mensajes de rastreo con la GUI de rastreo

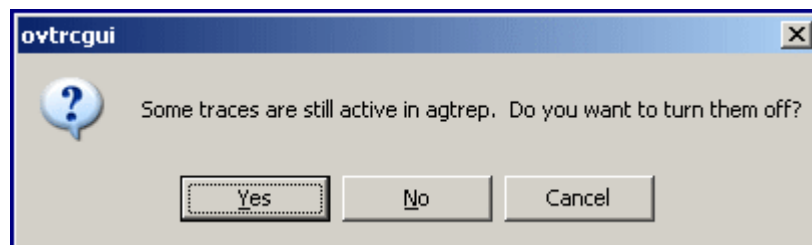
En los nodos de Windows, puede usar la utilidad `ovtrcgui` para configurar el rastreo y ver los mensajes del rastreo.

Habilitación del mecanismo de rastreo

Para habilitar el mecanismo de rastreo con la utilidad `ovtrcgui` y sin la ayuda de un archivo de configuración del rastreo, siga estos pasos:

- 1 Siga los pasos del [paso 1](#) en la página 412 al [paso 6](#) en la página 415 en [Uso de la GUI de rastreo](#) en la página 412.
- 2 Cierre el editor de configuración de rastreo.
- 3 Haga clic en **No** cuando se le solicite que guarden los cambios en `Untitled`.

Aparece el siguiente mensaje:



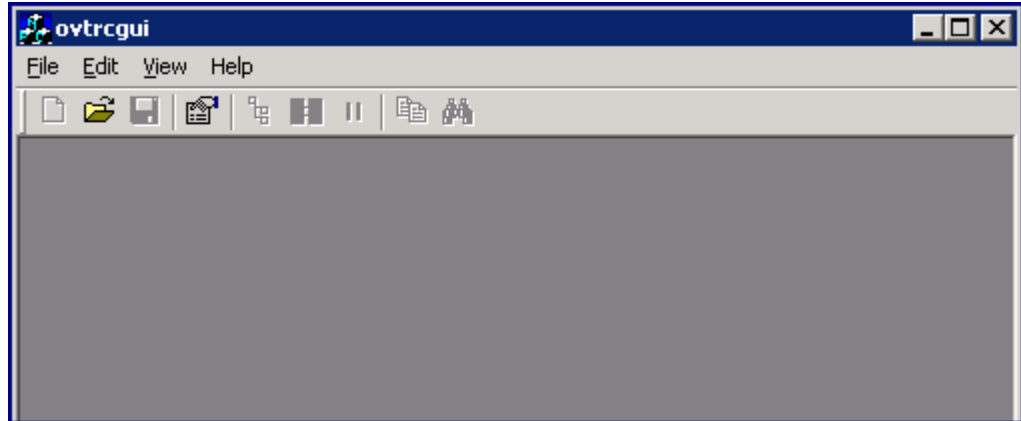
- 4 Haga clic en **No**. Si hace clic en **Yes**, la utilidad `ovtrcgui` deshabilita inmediatamente el mecanismo de rastreo.

Para habilitar el mecanismo de rastreo con la utilidad `ovtrcgui` usando un archivo de configuración del rastreo, vaya a la ubicación en el sistema local donde el archivo de configuración del rastreo esté disponible y haga doble clic en dicho archivo. Si lo prefiere, abra la utilidad `ovtrcgui`, haga clic en **File** → **Open**, seleccione el archivo de configuración del rastreo y haga clic en **Open**.

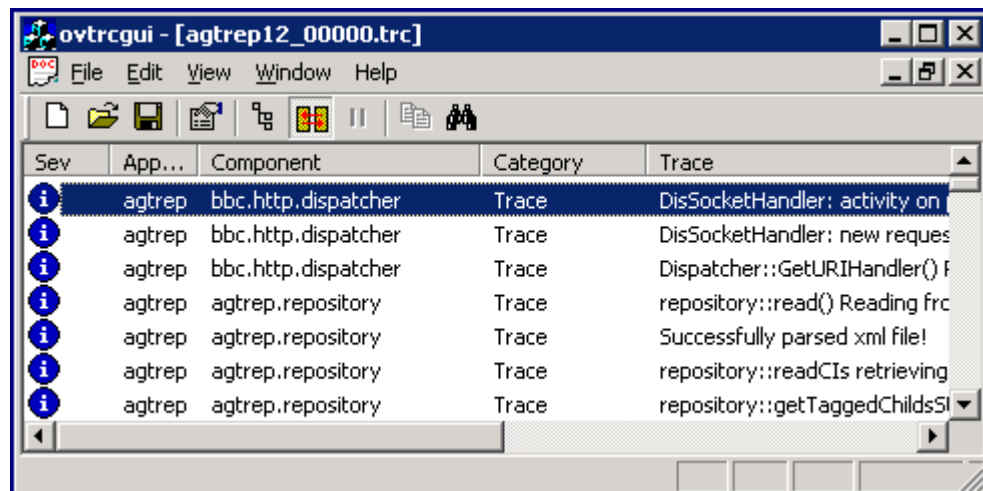
Visualización de mensajes de rastreo

Para ver los archivos de resultados del rastreo con la utilidad `ovtrcgui`, siga estos pasos:

- 1 Ejecute el archivo `ovtrcgui.exe` en el directorio `%OvInstallDir%\support`. Se abre la ventana `ovtrcgui`.

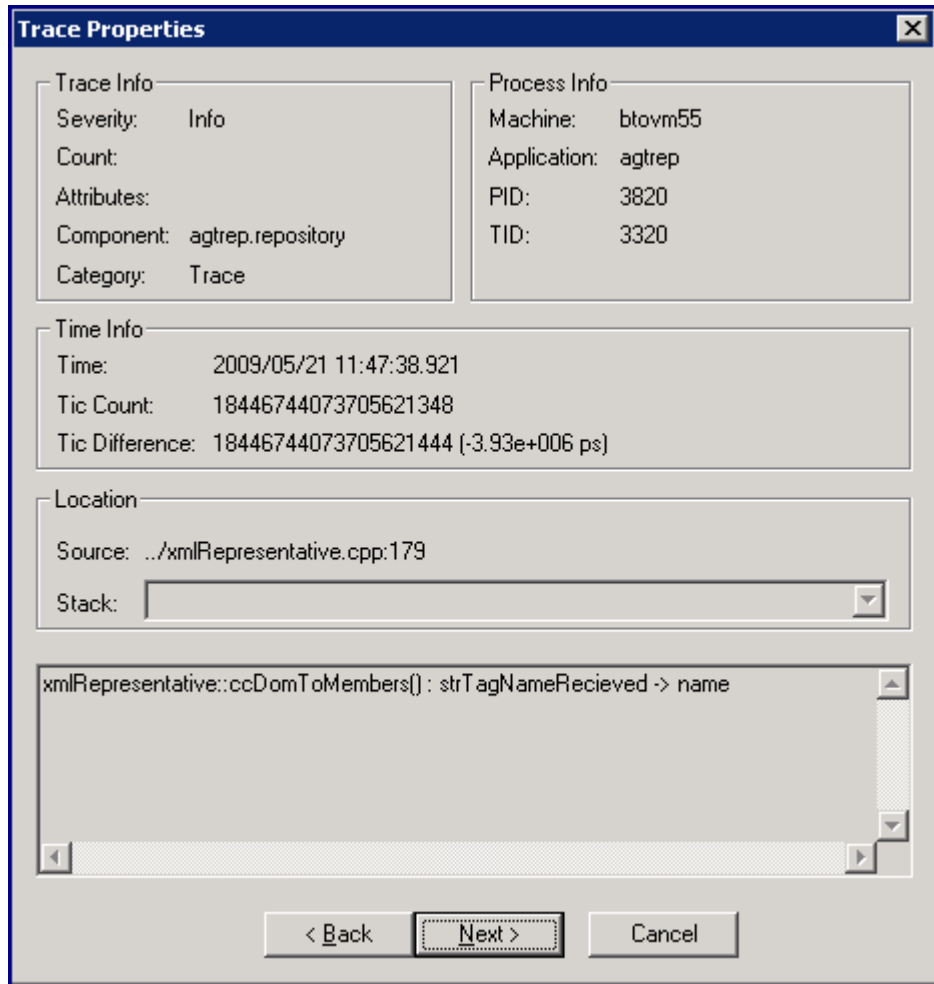


- 2 Haga clic en **File** → **Open**. Se abrirá el cuadro de diálogo **Open**.
- 3 Desplácese a la ubicación del archivo de resultados del rastreo, seleccione el archivo `.trc` y haga clic en **Open**. La utilidad `ovtrcgui` para muestra el contenido del archivo `.trc`.



Cada nueva línea en el archivo `.trc` representa un nuevo mensaje de rastreo.

- 4 Haga doble clic en un mensaje de rastreo para ver los detalles. Se abre la ventana Trace Properties.



La ventana Trace Properties presenta los siguientes detalles:

- Trace Info:
 - *Severity*: la gravedad del mensaje de rastreo.
 - *Count*: el número de serie del mensaje.
 - *Attributes*: el atributo del mensaje de rastreo.
 - *Component*: nombre del componente que emite el mensaje de rastreo.
 - *Category*: nombre arbitrario asignado por la aplicación rastreada.
- Process Info:
 - *Machine*: nombre del host del nodo.
 - *Application*: nombre de la aplicación rastreada.
 - *PID*: ID del proceso de la aplicación rastreada.
 - *TID*: ID del subprocesso de la aplicación rastreada.




- Time Info:
 - *Time*: fecha y hora equivalente local del mensaje de rastreo.
 - *Tic count*: tiempo transcurrido de alta resolución.
 - *Tic difference*:
 - Location
 - *Source*: número de línea y nombre de archivo del origen que genera el rastreo.
 - *Stack*: descripción de la pila de llamada en la aplicación rastreada.
- 5 Haga clic en **Next** para ver el siguiente mensaje de rastreo.
 - 6 Después de ver todos los mensajes de rastreo, haga clic en **Cancel**.

Uso de la vista de la lista de rastreos

De forma predeterminada, la utilidad `ovtrcgui` muestra los mensajes de rastreo de un archivo de rastreo en la vista de lista de rastreos. La vista de lista de rastreos presenta los mensajes de rastreo en formato de tabla.

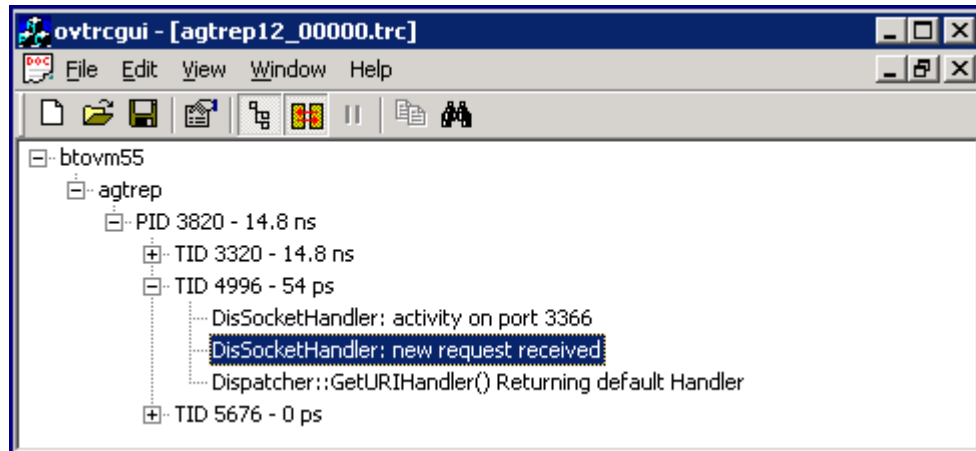
La vista de lista de rastreos presenta cada mensaje de rastreo con ayuda de las siguientes columnas:

Tabla 13 Vista de lista de rastreos

Columna	Descripción
Severity	Indica la gravedad del mensaje de rastreo. La vista usa los siguientes iconos para mostrar la gravedad de los mensajes:
	<ul style="list-style-type: none"> • Info 
	<ul style="list-style-type: none"> • Advertencia 
	<ul style="list-style-type: none"> • Error 
Application	Muestra el nombre de la aplicación rastreada.
Component	Muestra el nombre del componente de la aplicación rastreada que generó el mensaje de rastreo.
Category	Muestra la categoría del mensaje de rastreo.
Trace	Muestra el texto del mensaje de rastreo.

Uso de la vista de árbol de procedimientos

Puede ver los mensajes de rastreo en un formato estructurado en la vista de árbol de procedimientos. La vista de árbol de procedimientos ordena los mensajes en función de los Id. de proceso y los Id. de subprocesos y presenta los datos en forma de una vista de árbol.



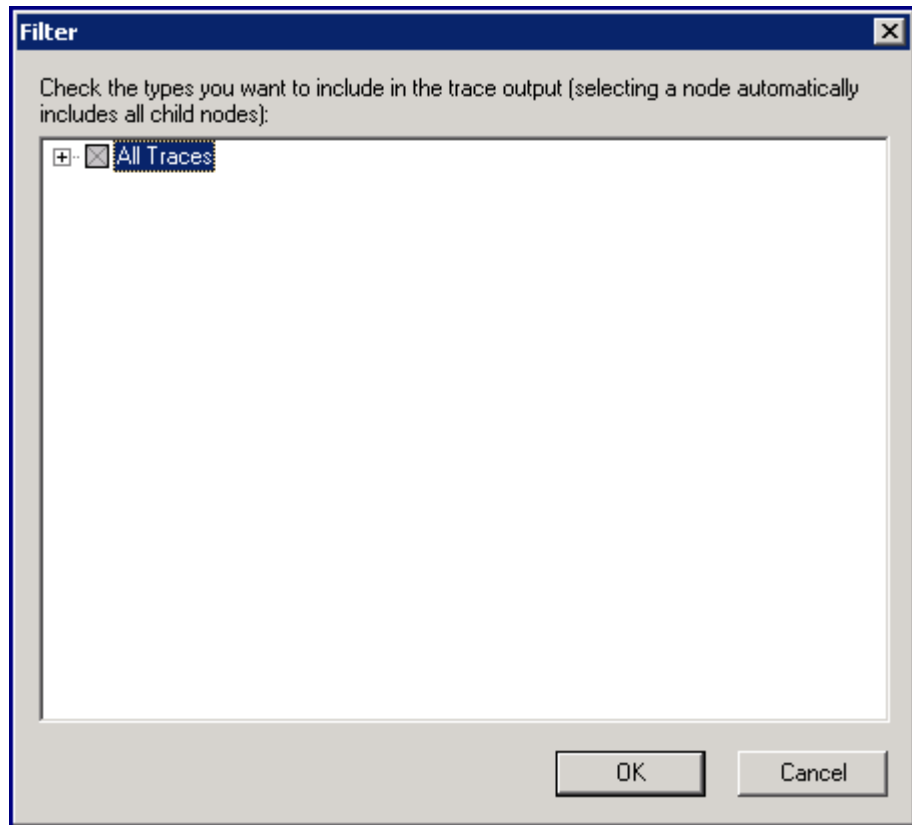
Puede ampliar los Id. de proceso y los Id. de subprocesos para ver los mensajes de rastreo.

Para volver a la vista de la lista de rastreo, haga clic en .

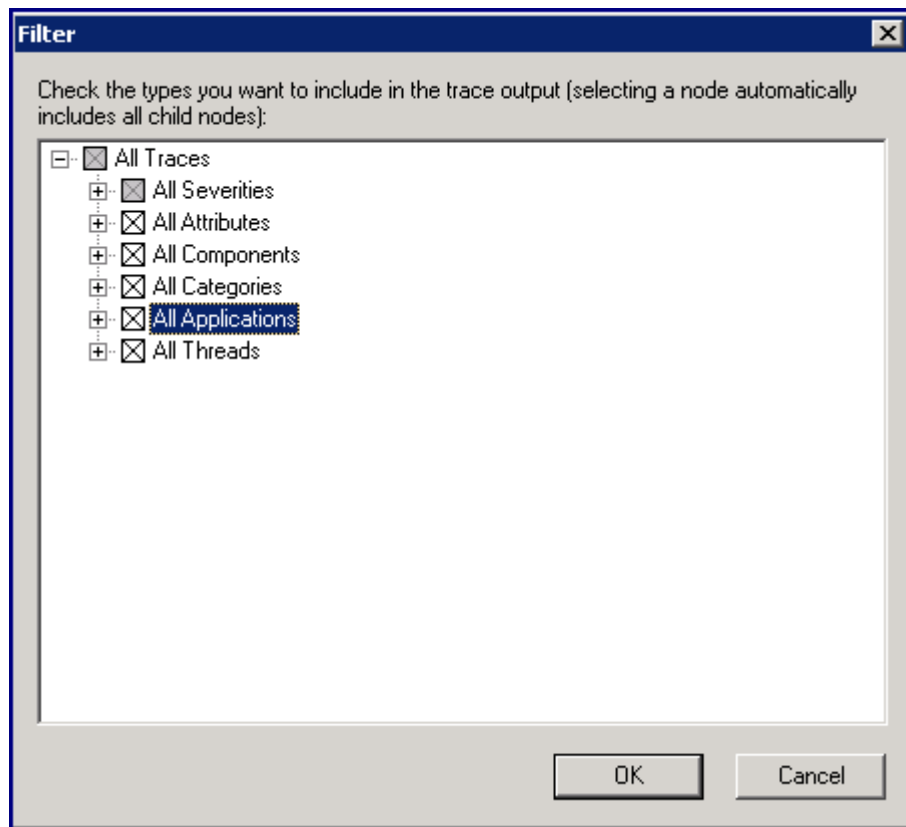
Filtrado de rastreos

La utilidad ovtregui muestra todos los mensajes de rastreo que están registrados en los archivos de resultados de rastreo basados en la configuración establecida en el archivo de configuración del rastreo. Puede filtrar los mensajes disponibles para mostrar sólo los mensajes que elija en la consola ovtregui. Para filtrar los mensajes de rastreo disponibles, siga estos pasos:

- 1 En la consola ovtregui, haga clic en **View** → **Filter**. Se abrirá el cuadro de diálogo Filter.



- Amplíe **All Traces**. En el cuadro de diálogo se enumeran todos los parámetros de filtrado en forma de árbol.



- Amplíe los parámetros para realizar selecciones para filtrar los mensajes de rastreo.
- Haga clic en **OK**. Sólo puede ver los mensajes filtrados en la consola ovtregui.

27 Solución de problemas

La presente sección describe las soluciones y medidas a tomar para resolver los problemas más comunes que pueden surgir al trabajar con HP Operations Agent 11.00. Entre las áreas cubiertas en esta sección se incluyen:

- Componente de monitorización de operaciones
- Componente Performance Collection
- RTMA

Componente de monitorización de operaciones

- *Problema:* En el nodo de Windows Server 2008, el proceso `opcmsga` no funciona y el comando `ovc` muestra el estado del proceso `opcmsga` como `aborted`.

Solución:

Establezca la variable `OPC_RPC_ONLY` en `TRUE` ejecutando el comando siguiente:

```
ovconfchg -ns eaagt -set OPC_RPC_ONLY TRUE
```

- *Problema:* En nodos de Windows, las secuencias Perl no funcionan en las directivas.

Causa: La secuencia Perl disponible en las directivas requiere la variable de configuración `PATH` para incluir el directorio en el que Perl (suministrado con HP Operations Agent) se encuentra disponible.

Solución:

- a Ejecute el comando siguiente para establecer la variable de configuración `PATH` en el directorio Perl:

```
ovconfchg -ns ctrl.env -set PATH "%ovinstalldir%nonOV\perl\bin"
```

- b Reinicie el agente ejecutando los siguientes comandos:

```
— ovc -kill  
— ovc -start
```

- *Problema:* No surten efecto los cambios tras cambiar los valores de variables con el comando `ovconfchg`.

Causa 1:

La variable requiere que se reinicie el agente.

Solución 1:

Reinicie el agente ejecutando los siguientes comandos:

- a **ovc -kill**
- b **ovc -start**

Causa 2:

Las directivas ConfigFile implementadas en el nodo establecen la variable en un valor específico.

Solución:

Si las directivas implementadas ConfigFile incluyen comandos para establecer las variables de configuración en valores específicos, no surtirán efecto los cambios realizados con el comando `ovconfchg`. Deberá o bien eliminar del nodo las directivas ConfigFile, o bien modificar las directivas para incluir los comandos que establecen las variables en los valores deseados.

Causa 3:

El perfil o archivo de trabajo disponible en el nodo invalida los cambios del usuario.

Solución:

Abra el perfil o archivo de trabajo del nodo y asegúrese de que no incluyan configuraciones conflictivas para las variables.

- *Problema:* Una vez que cambie el valor de la variable de configuración `SNMP_SESSION_MODE`, el estado del proceso `opctrapi` se mostrará como Aborted por `ovc`.

Causa:

Una vez que cambie el valor de la variable de configuración `SNMP_SESSION_MODE`, HP Operations Agent intentará reiniciar `opctrapi`. Ocasionalmente el proceso de reinicio de `opctrapi` puede no funcionar correctamente.

Solución:

Reinicie `opctrapi` ejecutando los siguientes comandos:

ovc -start opctrapi

Componente Performance Collection

- *Problema:* El siguiente error se muestra en el archivo `status.midaemon` en el sistema HP-UX 11.11:

```
mi_shared - MI initialization failed (status 28)
```

Causa: Tamaño de página grande del binario `midaemon`.

Solución: Para resolverlo, siga estos pasos:

- a Inicie una sesión en el sistema como usuario raíz.
- b Ejecute el comando siguiente para detener HP Operations Agent:
/opt/OV/bin/opcagt -stop
- c Ejecute el siguiente comando para hacer una copia de seguridad de `midaemon`:
cp /opt/perf/bin/midaemon /opt/perf/bin/midaemon.backup

- d Ejecute el siguiente comando para reducir el tamaño de página del binario `midaemon` a 4K:

```
chattr +pi 4K /opt/perf/bin/midaemon
```

- e Ejecute el comando siguiente para iniciar HP Operations Agent:

```
/opt/OV/bin/opcagt -start
```

- Tras instalar HP Operations Agent, aparece el siguiente mensaje de error en el archivo `System.txt` si se activa el mecanismo de rastreo:

Error en la inicialización del origen de datos Scope

Solución: Ignore este error.

RTMA

- *Problema:* En el nodo de vSphere Management Assistant (vMA), el proceso `rtmd` no funciona y el comando `ovc` muestra el estado del proceso `rtmd` como `aborted`.

Causa: El proceso `rtmd` no puede convertir el nombre de host del sistema en la dirección IP.

Solución:

- a Inicie sesión en el nodo con los privilegios raíz.
- b En el directorio `/etc`, abra el archivo `hosts` con un editor de texto.
- c Localice la línea en donde aparece el término `localhost`.
- d Suprima el signo de almohadilla `#` del comienzo de la línea.
- e Guarde el archivo.
- f Inicie todos los procesos ejecutando el siguiente comando:

```
ovc -restart
```

- *Problema:* La ventana Diagnostic de HP Performance Manager no puede acceder a los datos.

Causa: El proceso `rtmd` no está siendo ejecutado.

Solución: Para comprobar si el proceso `rtmd` está siendo ejecutado en el nodo de HP Operations Agent node, ejecute `ovc -status rtmd`. Para iniciar el proceso `rtmd`, ejecute `ovc -start rtmd`.

- *Problema:* El siguiente error se muestra en el archivo `status.perfd` en el sistema HP-UX 11.11:

```
mi_shared - MI initialization failed (status 28)
```

Causa: Tamaño de página grande del binario `perfd`.

Solución: Para resolverlo, siga estos pasos:

- a Inicie una sesión en el sistema como usuario raíz.
- b Ejecute el comando siguiente para detener HP Operations Agent:

```
/opt/OV/bin/opcagt -stop
```

- c Ejecute el siguiente comando para hacer una copia de seguridad de perfd:
cp /opt/perf/bin/perfd /opt/perf/bin/perfd.backup
- d Ejecute el siguiente comando para reducir el tamaño de página del binario perfd a 4K:
chattr +pi 4K /opt/perf/bin/perfd
- e Ejecute el comando siguiente para iniciar HP Operations Agent:
/opt/OV/bin/opcagt -start

Índice

- A**
- abreviaciones de comandos
 - extract, 119
 - utility, 73
- acceder a datos DSI, 304
- acceder a help
 - programa extract, 134
 - programa utility, 78
- acción, 296
- acciones compuestas en la instrucción ALARM, 172
- acciones locales
 - alarmas, 175
 - ejecutar, 163
- administrar datos DSI, 305
- administrar SLO, 349
- agdb, 162
- agdbserver, 162
- agente de registro, Application Response Measurement 2.0, 354
- agregar nuevas aplicaciones, 359
- agregar nuevas transacciones, ttd.conf, 359, 371
- agsysdb, 162
- alarma
 - generador, 296
 - procesar, 296
- alarmas
 - acciones locales, 163
 - configurar, 296
 - definir, 295
 - enviar mensajes a Operations Manager, 162
- alarmas por datos de transacciones
 - con GlancePlus, 372, 373
 - con Performance Agent, 373
 - con Performance Manager, 372
- alarmdef
 - cambios, 295
- alerta, 296
- almacenar, anexar datos, 237
- almacenar datos de archivos de registro, 44, 139, 152, 154, 237, 238
- almacenar procesos, administrar, 45
- análisis
 - archivos de registro, 164
- análisis a largo plazo, 348
- analizar
 - archivos de registro, 74
 - datos históricos de archivos de registro, 74, 164
- analizar archivos de registro, 239, 240
- analizar datos
 - con GlancePlus, 372
 - con Performance Manager, 372
- analizar datos históricos de archivos de registro, 239
- anexar datos almacenados, 237
- API de ARM
 - biblioteca compartida, 367
 - devolución de estado, 357
 - instrumentar scopeux, 386
 - llamadas de función, 350
 - mensajes de error de, 375
 - seguimiento de transacciones y, 353
- aplicaciones
 - agregar nuevas, 359
 - definidas en ttd.conf, 362
- Application Response Measurement
 - agente de registro de 2.0, 354
 - biblioteca (libarm), 388
 - consideraciones de sobrecarga, 364
 - directrices para usar, 364
 - ejemplo de aplicaciones, 355
 - funciones de 2.0, 354
 - kit de desarrollo de software (SDK) de 2.0, 354
 - obtener biblioteca compartida, 367
 - sin biblioteca de operaciones (libarmNOP), 395
 - ventajas, 348
- archivo alarmdef, 74, 76, 180, 186, 295
- archivo de configuración de transacciones, 359
- archivo de configuración de transacciones, consulte ttd.conf, 353

- archivo de exportación
 - atributos, 227, 230
 - formato ASCII, 227
 - formato binario, 227
 - formato de archivo de datos, 227
 - nombres de archivo predeterminados, 230
 - título, 109
 - WK1 (hoja de cálculo), formato, 227
- archivo de formato, 296, 302
- archivo de inclusión arm.h, 394
- archivo de plantilla de exportación
 - configurar, 235
 - crear una plantilla rápida, 233
 - diseño, 107
 - diseño múltiple, 229
 - elementos, 108
 - encabezados, 107, 229
 - formato, 106
 - formato de archivos, 227
 - minutos de resumen, 229
 - parámetros, 106
 - que falta, 107
 - report, 106
 - salida, 108
 - separador, 107
 - separador de campo, 229
 - sintaxis, 106
 - summary, 107
 - tipo de datos, 108
 - título de archivo de exportación, 109
 - valor que falta, 229
- archivo de registro
 - tamaño, control, 282
- archivo logappl, 24
 - grupos PRM, 24
- archivo logdev, 24
- archivo logglob, 23, 154
- archivo logproc, 24
- archivo parm
 - comprobación de sintaxis, 82
 - configurar, 245
 - configurar intervalos de registro de datos, 37
 - flush, 31
 - gapapp, 28
 - modificaciones de Performance Manager y Performance Agent, 369
 - modificar, 19, 246, 256
 - parámetros, 23
 - parámetros de definiciones de aplicación, 32
 - parámetro subprociinterval, 28
- archivo perflbd.mwc
 - formato, 250
- archivo reptall, 104
- archivo reptfile, 103, 144
- archivo reptfile.mwr, 230
- archivo repthist, 104
- archivos
 - alarmdef, 74, 76, 186, 295
 - definiciones de alarma, 74, 239, 247
 - logappl, 24
 - logdev, 24
 - logglob, 23
 - logproc, 24
 - parámetros de recopilación, 245
 - parm, 245
 - plantilla de exportación, 103
 - reptall, 104
 - reptfile, 103, 144
 - reptfile.mwr, 230
 - repthist, 104
 - status.scope, 17
 - ttdconf.mwc, 252
- archivos de plantilla de exportación personalizados, 104, 235
- archivos de registro
 - almacenar datos, 44
 - cambiar de tamaño, 84
 - controlar espacio en disco., 42
 - DSI, 130, 264
 - establecer el tamaño máximo, 29, 43
 - exploración, 88
 - organización, 269
 - revertir, 43, 44
- archivos de registro DSI, 130, 133
- archivos de registro sin formato
 - administrar espacio, 84
 - nombres, 80
- archivo status.scope, 17
- archivo ttdconf.mwc, 252
- argumentos de línea de comandos
 - programa extract, 98
 - programa utility, 62
- asignar datos de entrada en especificación, 302
- asignar nombre a una transacción, 360, 370
- atributos de archivo, exportar, 227

B

barra de estado, visualizar, 243
barra de herramientas, visualizar, 243
base de datos agdb, 162
bibliotecas
 usar libarm, 367
bibliotecas compartidas, 395

C

cambiar
 archivo alarmdef, 295
 especificaciones de clase, 303
cambiar de tamaño
 archivos de registro, 84
cambiar el tamaño del segmento de memoria
 compartida de midaemon, 370
cambiar range o SLO, ttd.conf, 360, 371
cambiar tamaño
 tareas, 44
cambiar tamaño de archivos de registro, 241, 242
canalización con nombre, 296
canalizar datos a dsilog, 296
capacidad, 282
capturas SNMP, 296
caracteres de escape, 276, 277, 287
clase
 capacidad, 282
 definiciones, 273
 descripción, 269
 enumerar con sdlutil, 305
 etiqueta, 276
 instrucción, 275
 intervalo de índice, 276
 intervalo de supresión, 277
 parámetro capacity, 284
 registros por hora, 283
 requisitos de ID, 275
 requisitos de nombre, 275
 sintaxis, 275
 valores predeterminados de descripciones, 275
comando analyze, programa utility, 74
comando application, programa extract, 123
comando checkdef, programa utility, 76
comando class, programa extract, 124
comando configuration, programa extract, 125
comando cpu, programa extract, 125
comando detail, programa utility, 77

comando disk, programa extract, 126
comando exit, programa extract, 127
comando exit, programa utility, 78
comando export, programa extract, 102, 127
comando extract, programa extract, 130
comando filesystem, programa extract, 131
comando global, programa extract, 132
comando guide, programa extract, 133
comando guide, programa utility, 78
comando help, programa extract, 134
comando help, programa utility, 78
comando list, programa extract, 134
comando list, programa utility, 79
comando logfile, programa utility, 80
comando lvolume, programa extract, 137
comando menu
 programa extract, 137
 programa utility, 82
comando monthly, programa extract, 139
comando output, programa extract, 141
comando parmfile, programa utility, 82
comando perfstat, 17
comando process, programa extract, 142
comando quit
 programa extract, 143
 programa utility, 83
comando report, programa extract, 144
comando resize
 informes, 86
 parámetros de cambio de tamaño
 predeterminados, 85
 programa utility, 61, 84
comandos
 perfstat, 17
 programa extract, 119
 programa utility, 73
comandos del programa utility
 resize, 61

- comandos extract
 - application, 123
 - class, 124
 - configuration, 125
 - cpu, 125
 - disk, 126
 - exit, 127
 - export, 102, 127
 - extract, 130
 - filesystem, 131
 - global, 132
 - guide, 133
 - help, 134
 - list, 134
 - lvolume, 137
 - menu, 137
 - monthly, 139
 - output, 141
 - process, 142
 - quit, 143
 - report, 144
 - sh, 144
 - shift, 145
 - show, 146
 - start, 147
 - stop, 149
 - weekdays, 151
 - weekly, 152
 - yearly, 154
- comando sh
 - programa extract, 144
 - programa utility, 89
- comando shift, programa extract, 145
- comando show
 - programa extract, 146
 - programa utility, 90
- comando start
 - parámetros, 92
 - programa extract, 147
 - programa utility, 91
- comando stop
 - parámetros, 93
 - programa extract, 149
 - programa utility, 93
- comandos utility
 - analyze, 74
 - checkdef, 76
 - detail, 77
 - exit, 78
 - guide, 78
 - help, 78
 - list, 79
 - logfile, 80
 - menu, 82
 - parmfile, 82
 - quit, 83
 - resize, 84
 - scan, 88
 - sh, 89
 - show, 90
 - start, 91
 - stop, 93
- comando weekdays, programa extract, 151
- comando weekly, programa extract, 152
- comando yearly, programa extract, 154
- comentarios, uso en sintaxis de alarma, 167
- command scan, programa utility, 88
- compatibilidad
 - Application Response Measurement 2.0, 354
- compilador sdlcomp, 268
- compilar la especificación de clase, 309, 313, 320
- componentes del seguimiento de transacciones, 353
- comprobar estado de Performance Agent, 256
- condiciones
 - en sintaxis de alarma, 171
 - sintaxis de alarma, 168, 177
- condiciones de alarma en datos históricos de archivo de registro, 74, 239, 240
- condiciones de alarma en datos históricos de archivos de registro, 164
- condiciones de desbordamiento, 370
- Configuración de intervalos de registro de datos, 37
- configurar
 - agente de monitorización, 47
 - archivo de definiciones de alarma, 247
 - archivo parm, 245
 - archivos de plantilla de exportación, 235
 - archivo ttdconf.mwc, 252
 - cadenas de comunidad, 48
 - parámetros de recopilación, 245
 - transacciones, 252
- configurar alarmas, 296
- configurar opciones de usuario, 243

- configurar rastreo, 405
- conjuntos de archivos de registro
 - asignar nombre, 273
 - definir, 269
 - enumerar con sdlutil, 305
 - suprimir, 282
- consideraciones de control de errores, 369
- constantes, en sintaxis de alarma, 168
- contadores de rendimiento, crear recopilaciones de, 258
- Contenedores Java, 395
- contenedores Java
 - actualizar los datos de instancia de transacción, 399
 - configurar una aplicación, 396
 - configurar una transacción, 396
 - detener una instancia de transacción, 400
 - documentación, 402
 - ejemplos, 395
 - iniciar instancia de transacción, 398
 - usar transacción completada, 401
- control del espacio en disco usado por los archivos de registro, 42
- convenciones, sintaxis de alarma, 166
- correladores ARM, 364
- crear
 - archivos de registro, 269
 - especificación de clase, 264
- crear gráficos o informes personalizados, 109
- crear recopilaciones de contadores de rendimiento, 258
- crear una plantilla de exportación rápida, 233

D

- datos
 - acceder, 304
 - administrar, 305
 - exportar, 304
 - recopilar, 265
 - registrar, 265
- datos de archivos de registro
 - almacenar, 139, 152, 154, 237, 238
 - analizar en busca de condiciones de alarma, 164, 239, 240
 - cambiar tamaño, 241, 242
 - explorar, 240, 241
 - exportar, 127, 227, 232
 - extraer, 130, 226
- datos de entrada en dsilog, 296

- datos registrados, exportar, 304
- default ttd.conf, archivo, 359, 360, 362
- definición de alarma
 - nombre de métrica DSI, 295
- definiciones de alarma
 - archivo, 74, 239, 247
 - componentes, 165
 - comprobar sintaxis, 76
 - configurar, 247
 - ejemplos, 184
 - métrica de aplicaciones, 168
 - modificar, 248
 - nombres de métrica, 168
 - personalizar, 186
- definir
 - objetivos de nivel de servicio, 368, 370
 - rangos de medición, 370
- delimitadores, 288, 299
- demonio de interfaz de medición, consulte midaemon, 353
- demonio de registro de seguimiento de transacciones, consulte ttd, 353
- detener
 - aplicaciones, 357
 - recopilación de datos, 41
 - scopeux, 41
 - ttd, 357
- detener y reiniciar scopeux, 357, 360
- directrices para usar ARM, 364
- diseño de registros de encabezado binarios, 112
- diseño múltiple, especificar en archivos de exportación, 229
- DSI. Véase también integración de orígenes de datos
- dsilog
 - entradas, 296
 - escribir una secuencia de comandos, 308
 - proceso de registro, 296, 310
 - sintaxis, 296

E

- E/S de disco, sobrecarga, 364
- ejecución
 - programa extract, 96
- ejecutar
 - programa utility, 60
 - ttd, 357
 - una aplicación, 369
- ejecutar acciones locales, 163
- ejecutar una aplicación, 369

- ejemplo de aplicación, 349
- ejemplo de aplicaciones instrumentadas para ARM, 355
- ejemplo de recuento de palabras who, 327
- ejemplos
 - seguimiento de transacciones, 379
 - ttd.conf, 381
- Ejemplos de la opción de compilador C por plataforma, 394
- ejemplos del uso de DSI, 307
 - escribir una secuencia de comandos dsilog, 308
 - registrar datos sar desde un archivo, 311
 - registrar datos sar desde varios archivos, 315
 - registrar datos sar para varias opciones, 321
 - registrar datos vmstat, 309
 - registrar el número de usuarios del sistema, 327
- encabezados de columna, especificar en archivos de exportación, 229
- enviar capturas SNMP, 162
- enviar información de alarma, 296
- enviar mensajes de alarma, 162, 174
- errores, procesamiento de alarmas, 164
- errores de procesamiento de alarmas, 164
- escribir una secuencia de comandos dsilog, 308
 - ejemplo de secuencia de comandos dsilog problemática, 308
 - ejemplo de secuencia de comandos dsilog recomendada, 308
- espacio en disco usado por los archivos de registro, controlar, 42
- especificación de clase
 - cambiar, 303
 - compilar, 309, 313, 320
 - crear, 309, 312, 315
 - definición de métricas, 285
 - pruebas, 299
 - recrear con sdlutil, 305
- establecer el tamaño máximo de los archivos de registro, 43
- estadísticas, enumerar con sdlutil, 305
- etiqueta
 - clase, 276
 - métrica, 286
- examinar tendencias, 348
- excluir datos del registro, 302
- explorar datos de transacciones con Performance Agent, 371
- explorar un archivo de registro, 88
- exportar archivos de salida predeterminados, 128
- exportar datos de archivos de registro, 127, 227, 232
 - en función de fechas y horas, 225
- exportar datos de archivos de registro DSI, 133
- exportar datos de transacciones con Performance Agent, 371
- exportar datos registrados, 304, 311
- exportar tipos de datos, 103
- expresiones, en sintaxis de alarma, 168
- Extended Collection Builder and Manager, 258
 - sugerencias para usar, 258
- extraer
 - usar con datos de transacciones, 370
- extraer datos de archivos de registro, 130, 226
 - en función de fechas y horas, 225
- extraer datos de transacciones con Performance Agent, 371

F

- fifo, 296
- finalización
 - programa extract, 127, 143
- finalizar
 - comandos utility, 83
 - programa utility, 78
- Flush, 31
- formato ASCII, archivo de exportación, 106, 227
- formato binario, archivo de exportación, 106, 227
- formato de archivo de datos, archivo de exportación, 106, 227
- formato de registro ASCII, 110
- formato de registro binario, 111
- formato de registro de archivo de datos, 110
- formato de WK1 (hoja de cálculo), archivo de exportación, 227
- formatos de registro
 - archivo de datos, 110
 - ASCII, 110
 - binario, 111
- formato WK1, archivo de exportación, 106
- función C
 - arm_stop, 400

- función export
 - archivos de datos, 104
 - archivos de plantilla de exportación, 103
 - introducción general, 102
 - proceso, 102
 - sintaxis de archivo de plantilla de exportación, 106
 - tareas de ejemplo, 104
 - uso, 109

G

- gapapp, 28
- generador de alarmas, 162
- GlancePlus
 - alarmas por datos de transacciones, 372, 373
 - analizar datos de transacciones, 372
 - compatibilidad de Application Response Measurement 2.0, 354
 - identificar cuellos de botella en el rendimiento, 351
 - monitorizar datos de transacciones, 372
 - ver datos de transacciones, 351
- grupos PRM
 - APP_NAME_PRM_GROUPNAME, 24

I

- identificar cuellos de botella en el rendimiento, 351
- implementar una aplicación, 369
- información de versión, presentar, 305
- información sobre herramientas, visualizar, 243
- informar de condiciones de alarma en datos históricos de archivo de registro, 239
- informar del contenido del archivo de registro, 240, 241

- informe de exploración de utility
 - Definiciones de aplicación iniciales del archivo parm, 66
 - información global inicial del archivo parm, 65
 - informe de resumen específico de la aplicación, 68
 - Notificaciones de adiciones/eliminaciones en el archivo parm, 67
 - notificaciones de cambios globales del archivo parm, 67
 - Notificaciones de tiempo de inactividad de scopeux, 67
 - parada e inicio de exploración, 70
 - resumen de cobertura del recopilador, 70
 - resumen de contenidos del archivo de registro, 71
 - resumen de espacio libre del archivo de registro, 72
 - resumen de motivo de registro de proceso, 69
 - resumen general de aplicación, 70
- iniciar proceso de registro, 296
- instrucción ALARM, sintaxis de alarma, 170
- instrucción ALERT, sintaxis de alarma, 174
- instrucción ALIAS, sintaxis de alarma, 182
- instrucción application LOOP, sintaxis de alarma, 178
- instrucciones compuestas en sintaxis de alarma, 167
- instrucción EXEC, sintaxis de alarma, 174
- instrucción IF, sintaxis de alarma, 177
- instrucción INCLUDE, sintaxis de alarma, 179
- instrucción LOOP, sintaxis de alarma, 178
- instrucción PRINT, sintaxis de alarma, 176
- instrucción SYMPTOM, sintaxis de alarma, 183
- instrucción USE, sintaxis de alarma, 180
- instrucción VAR, sintaxis de alarma, 182
- integración de orígenes de datos
 - ejemplos del uso de DSI, 307
 - funcionamiento, 264
 - introducción general, 263
 - mensajes de error, 329
 - pruebas, 299
- interfaz de línea de comandos
 - programa extract, 96, 97
 - programa utility, 60, 62
- intervalo de índice, clase, 276
- introducción al seguimiento de transacciones, 353
- introducción general
 - integración de orígenes de datos, 263

- L**
- libarm, 367, 388
- libarmNOP, 395
- límites en transacciones exclusivas, 370
- llamadas a la API de ARM
 - arm_complete_transaction, 355
- llamadas API ARM
 - arm_getid call, 396
 - arm_init call, 396
 - arm_start call, 398
- longitud de métrica de texto, 288
- M**
- marca de hora, 286
 - eliminar, 296
- marca de hora UNIX, 286
- medición, definir rangos, 370
- mensajes de error, 329
 - de la API de ARM, 375
 - de midaemon, 375
- mensajes en sintaxis de alarma, 169
- método de resumen, 287
- métrica, 377
 - definición, 285
 - descripción, 269
 - enumerar con sdlutil, 305
 - etiqueta, 286
 - método de resumen, 287
 - orden, 286
 - palabra clave, 285
 - precision, 288
 - requisitos de etiqueta, 287
 - requisitos de ID, 285
 - requisitos de nombre, 285
 - reutilización de nombre, 286
 - reutilizar nombre, 286
 - texto, 288
 - valores predeterminados, 286
- métrica de aplicaciones en definiciones de alarmas, 168
- métrica de texto
 - archivo de formato, 302
 - especificar, 288
- métrica DSI en definiciones de alarmas, 295
- métrica en definiciones de alarmas, 295
- métrica numérica, archivo de formato, 302
- métricas, seleccionar para exportación, 234, 236
- métricas definidas por el usuario, 384
- métricas de transacciones, 377
- MIB ID, 49
- midaemon, 353, 369
 - cambiar el tamaño del segmento de memoria compartida de midaemon, 370
 - errores, 357
 - mensajes de error, 375
 - segmento de memoria compartida, 357, 369
 - sobrecarga de la memoria, 365
- minutos de resumen, especificar en archivos de exportación, 229
- modificar
 - archivo parm, 19, 246, 256
 - definiciones de alarma, 248
 - parámetros de recopilación, 19, 246, 256
- modificar archivo de especificación de clase, 303
- modificar el archivo parm, 369
- modo de registro de aplicación PRM, 32
- modo guiado
 - extract, 133
 - utility, 78
- modo interactivo
 - programa extract, 97
 - programa utility, 61
- monitorizar datos de rendimiento de transacciones, 350
- N**
- Network Node Manager, 296
- Network Node Manager de HP, 162
- nivel de resumen, 296
 - predeterminado, 283
- niveles de resumen, 224
- nombres de archivo, archivos de exportación predeterminados, 230
- nombres de archivo de exportación predeterminados, 230
- nombres de métrica en sintaxis de alarma, 168, 183
- nombres de transacciones, 360
- O**
- objetivos de nivel de servicio
 - administrar, 349
 - definir, 368
- OID, 48
- opción cpu, 26
- opción de formato numérico, 302

- opción disk, 26
- opciones de usuario, configurar, 243
- opción memory, 26
- opción nokilled, 26
- opción shortlived, 26
- Operations Manager, 162, 296
- orden de métrica, cambio, 302
- Origen de datos predeterminado SCOPE, 168
- origen de datos predeterminado SCOPE, 180
- orígenes de datos, 180
- ovpa restart, 369
- ovpa start, 369
- ovpa stop scope, 368

P

- palabra clave argv1, archivo parm, 34
- palabras clave
 - range, 361, 370
 - slo, 361, 370
 - tran, 360, 370
- palabras clave de aplicación del archivo parm
 - argv1, 34
- parámetro
 - subprocinterval, 28
- parámetro application name, archivo parm, 33
- parámetro capacity
 - instrucción, 284
- parámetro cmd, archivo parm, 35
- parámetro de definiciones de aplicación, archivo parm, 32
- parámetro de diseño, archivo de plantilla de exportación, 107
- parámetro de diseño múltiple, archivo de plantilla de exportación, 229
- parámetro de elementos, archivo de plantilla de exportación, 108
- parámetro de encabezados, archivo de plantilla de exportación, 107, 229
- parámetro de formato
 - archivo de plantilla de exportación, 106
- parámetro de formato de archivos, archivo de plantilla de exportación, 227
- parámetro de informe, archivo de plantilla de exportación, 106
- parámetro de resumen, archivo de plantilla de exportación, 107, 229

- parámetro de salida, archivo de plantilla de exportación, 108
- parámetro de separación de campos, archivo de plantilla de exportación, 229
- parámetro de tipo de datos, archivo de plantilla de exportación, 108
- parámetro file, archivo parm, 33
- parámetro group, archivo parm, 36
- parámetro ID
 - archivo parm, 23
- parámetro javaarg, archivo parm, 31
- parámetro log, archivo parm, 23
- parámetro log predeterminado, archivo parm, 369
- parámetro mainttime, archivo parm, 30, 43
- parámetro or, archivo parm, 36
- parámetro priority, archivo parm, 36
- parámetro que falta, archivo de plantilla de exportación, 107, 229
- parámetro scopetransactions, archivo parm, 28
- parámetros de aplicación del archivo parm
 - cmd, 35
- parámetros del archivo parm
 - application name, 33
 - file, 33
 - group, 36
 - ID, 23
 - javaarg, 31
 - log, 23
 - mainttime, 30, 43
 - or, 36
 - priority, 36
 - scopetransactions, 28
 - size, 29
- parámetros de núcleo, 296
- parámetros de núcleo UNIX, 296
- parámetros de recopilación
 - configurar, 245
 - modificar, 246, 256
- parámetro separador, archivo de plantilla de exportación, 107
- parámetro size, archivo parm, 29
- parámetro threshold, archivo parm
 - opción cpu, 26
 - opción disk, 26
 - opción memory, 26
 - opción nokilled, 26
 - opción nonew, 26
 - opción shortlived, 26

- perfalarm, 162, 180
- Performance Agent
 - compatibilidad de Application Response Measurement 2.0, 354
 - comprobar estado, 256
 - exportar datos de transacciones, 371
 - extraer datos de transacciones, 371
 - modificar el archivo parm, 369
 - niveles de resumen, 224
 - programa extract, 95
 - programa utility, 59
 - recopilar y registrar datos, 371
 - reiniciar, 369
 - starting, 369
 - tipos de datos, 224
 - ver datos de transacciones, 350
- Performance Manager
 - alarmas por datos de transacciones, 372, 373
 - analizar datos de transacciones, 372
 - Mostrar datos DSI, 304
 - ver datos de transacciones, 350
- períodos de almacenamiento, 237
- persistencia del estado, 49
- personalizar el archivo ttd.conf, 370
- posiciones decimales, métrica, 288
- precision, 288
 - métrica, 288
- proccmd, 32
- procesar alarmas, 296
- proceso de registro, 296, 310
 - dsilog, 310
 - pruebas, 299
- procesos interesantes, 24, 43
- programa dsilog, 271
- programa extract, 95, 304
 - argumentos de línea de comandos, 98
 - comandos, 119
 - ejecución, 96
 - interfaz de línea de comandos, 97
 - modo interactivo frente a modo por lotes, 96
- programa utility, 59, 73, 164
 - argumentos de línea de comandos, 62
 - ejecutar, 60
 - ejemplo de modo por lotes, 61
 - ejemplo de programa interactivo, 61
 - interfaz de línea de comandos, 60, 62
 - introducir comandos shell, 89
 - modo interactivo, 61
 - modo interactivo frente a modo por lotes, 60
 - modo por lotes, 60

- pruebas
 - especificación de clase, 299
 - proceso de registro, 299

R

- range, palabra clave, 361
- rangos de datos para extraer o exportar, 225
- rastrear
 - archivos de configuración
 - aplicación, 408
 - rastrear, 409
 - sink, 408
 - versión de sintaxis, 407
 - configurar, 405
 - dinámico
 - habilitar, 416
- recopilación de datos
 - administración, 42
 - detener, 41
- recopilar datos con Performance Agent, 371
- recopilar datos del correlador, 369
- recopilar datos de recursos, 369
- registrar datos
 - ejecutar programa dsilog, 271
- registrar datos de transacciones, 350
- registro de datos variables, 37
- registro de nombre de aplicación, 116
- registro de nombre de dispositivo de disco, 117
- registro de nombre de netif, 117
- registro de nombre de transacción, 116
- registro de nombre de volumen lógico, 117
- registros por hora, 283, 296
- resumido por opción, 287
- reutilizar nombres de métrica, 286
- revertir archivos de registro, 44

S

- salida de compilador, ejemplo, 293
- salida de compilador de ejemplo, 293
- sar
 - ejemplo de registro de datos sar desde un archivo, 311
 - ejemplo de registro de datos sar desde varios archivos, 315
 - ejemplo de registro de datos sar para varias opciones, 321

- scopeux, 264
 - detener, 41
 - detener y reiniciar, 357, 360
 - instrumentar con llamadas a la API de ARM, 386
- SDL
 - prefijo para mensajes de error de especificación de clase, 291
- sdlcomp, 309
 - compilador, 309
- sdlgendata, 299
- sdlutil, 305, 311
 - sintaxis, 305
- secuencia de comandos mwa, 41
- segmento de memoria compartida, midaemon, 357, 369
- seguimiento de transacciones
 - componentes, 353
 - configurar una aplicación, 367
 - controlar errores, 369
 - datos que faltan, 370
 - ejemplos, 379
 - iniciar, 357
 - introducción general, 349
 - límites en transacciones exclusivas, 370
 - referencia técnica, 353
 - ventajas, 348
 - ver datos, 350
- separador, 288
- separadores, 299
- separar métricas en archivos de exportación, 229
- sin biblioteca de operaciones, 395
- sintaxis
 - dsilog, 296
 - exportar, 304
 - sdlutil, 305
- sintaxis de alarma, 166
 - comentarios, 167
 - condiciones, 168, 171, 177
 - constantes, 168
 - convenciones, 166
 - elementos comunes, 167
 - expresiones, 168
 - Instrucción ALARM, 170
 - instrucción ALERT, 174
 - instrucción ALIAS, 182
 - instrucciones compuestas, 167
 - Instrucción EXEC, 174
 - instrucción IF, 177
 - instrucción INCLUDE, 179
 - instrucción LOOP, 178
 - instrucción PRINT, 176
 - instrucción SYMPTOM, 183
 - instrucción USE, 180
 - instrucción VAR, 182
 - mensajes, 169
 - nombres de métrica, 168
 - referencia, 166
 - variables, 182
- SLO
 - Consulte objetivos de nivel de servicio, 349
- slo, palabra clave, 361
- SNMP
 - capturas, 162
 - nodos, 162
- SNMP_COMMUNITY, 48
- SNMP_COMMUNITY_LIST, 48
- sobrecarga
 - consideraciones para usar ARM, 364
 - CPU, 365
 - E/S de disco, 364
 - memory, 365
- sobrecarga de la CPU, 365
- sobrecarga de la memoria, 365
- solución de problemas sdlcomp, 294
- starting
 - Performance Agent, 369
 - ttd, 357
- status.mi, 375
- sugerencias de almacenamiento, 238
- supresión
 - acción, 277
 - ejemplo de acción, 278
 - intervalo, 277

T

tiempo de mantenimiento, archivo parm, 30

Tip of the Day, 243

tipos de datos, 103, 224, 383

título de archivo de exportación, 229

tran, palabra clave, 360

transacción

- agregar nueva a ttd.conf, 359

- asignar nombre, 370

- datos, 348

- métrica, 377

transacciones, configurar, 252

Transaction Tracker

- instrumentar una aplicación, 350

ttd, 353, 369

ttd.conf, 353, 359

- agregar nuevas transacciones, 359

- agregar transacciones, 371

- cambiar range o SLO, 360, 371

- ejemplo, 381

- formato, 362

- palabras clave, 360

- personalizar, 370

- predeterminado, 359, 360, 362, 370

U

utilidades, sdlutil, 305

V

valores predeterminados

- delimitadores, 288, 299

- Descripción de clase, 275

- etiqueta de clase, 276

- métrica, 286

- nivel de resumen, 283, 296

- registros por hora, 283

- separador, 288

- separadores, 299

variables, sintaxis de alarma, 182

ver datos de transacciones

- con GlancePlus, 351

- con Performance Agent, 350

- con Performance Manager, 350

- introducción general, 350

ver datos en Performance Manager, 304

vmstat

- ejemplo de registro de datos vmstat, 309

Z

zone_app, 31