Hewlett Packard Enterprise

HPE Operations Agent

Versión de software: 12.01
Para los sistemas operativos Windows®, HP-UX, Solaris, Linux y AIX

Guía de usuario

Fecha de publicación del documento: Agosto de 2017
Fecha de publicación del software: Agosto de 2017

Avisos legales

Garantía

Las únicas garantías para los productos y servicios de Hewlett Packard Enterprise Development Company, L.P se establecen en los términos de garantía expresos que acompañan a dichos productos y servicios. Nada de lo contenido en el presente documento podrá interpretarse como garantía adicional. HPE 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 HPE para su posesión, uso o copia. De conformidad con FAR 12.211 y 12.212, se autoriza el uso del software informático comercial, de la documentación del software informático y de los datos técnicos para componentes comerciales al gobierno de los EE.UU. bajo licencia comercial estándar del fabricante.

Aviso de copyright

Avisos de marcas registradas

Adobe® es una marca comercial de Adobe Systems Incorporated.

Microsoft® y Windows® son marcas comerciales registradas en EE.UU. del grupo de compañías de Microsoft.

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

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: **https://softwaresupport.hpe.com**

Este sitio requiere que el usuario se registre para obtener un HPE Passport y que inicie sesión. Para registrarse y obtener un ID de HPE Passport, vaya a: **https://hpp12.passport.hpe.com/hppcf/createuser.do**

O haga clic en el vínculo de **registro** en la parte superior de la página Soporte técnico de HPE Software.

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

Soporte

Visite el sitio web de Soporte técnico de HPE Software en: **https://softwaresupport.hpe.com**

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

El soporte técnico en línea de HPE 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 HPE
- 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 HPE Passport y que inicie sesión. Algunas pueden requerir también un contrato de soporte técnico. Para registrarse y obtener un ID de HPE Passport, vaya a:

https://hpp12.passport.hpe.com/hppcf/createuser.do

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

https://softwaresupport.hpe.com/web/softwaresupport/access-levels

HPE Software Solutions Now permite acceder al sitio web del portal de integración y soluciones de HPSW. Este sitio le permite explorar las soluciones de productos de HPE para satisfacer las necesidades de su negocio e incluye una lista completa de integraciones entre los distintos productos de HPE, así como una enumeración de los procesos de ITIL. La dirección URL de este sitio web es **https://softwaresupport.hpe.com/**

Contenido

Parte I: Introducción 15
Convenciones usadas en este documento 16

Parte II: Configuration 18
Capítulo 1: Cómo trabajar con HPE Operations Agent 19
Configuración del Monitor Agent 19
Configuración del agente de para monitorizar objetos MIB 20
Persistencia de un objeto monitorizado 21
Mejora de los parámetros de seguridad para ejecutar SNMPv3 GET 22
Habilitación de opcmona para ejecutar SNMPv3Get 23
Uso de la API SourceEX para agregar parámetros de seguridad a la directiva 25
Configuración del Componente Performance Collection de manera remota 28
Antes de comenzar 28
Implementación de la directiva OA-PerfCollComp-opcmsg 29
Configuración del Componente Performance Collection 30
Configuración del archivo parm 30
En HP Operations Manager para Windows 30
En HP Operations Manager en UNIX/Linux 9.10 31
Configuración del archivo alarmdef 32
En HP Operations Manager para Windows 32
En HP Operations Manager en UNIX/Linux 9.10 33
Trabajar de manera remota con HPE Operations Agent 34
Configuración del interceptor de capturas de SNMP 36
Configuración del interceptor de capturas de SNMP para capturas de SNMPv3 38
Configuración de opctrapi para interceptar capturas de SNMPv3 39
Cifrado de contraseñas con la utilidad opcpwcrpt 41
Mejora del interceptor de capturas de SNMP para interceptar capturas basadas en el Id. de objeto del enlace de variables 42
Habilitar opctrapi para usar el registro de NETSNMP 43
Integración de HPE Operations Agent con NNMi 45
Integración con la interfaz Northbound de NNMi para asociar la gravedad de los mensajes. 46
Configuración del interceptor de capturas de SNMP para mejorar la gravedad de los mensajes 46
Integración con la interfaz Northbound para agregar CMA a la consola de HPOM. 47
Configuración del interceptor de capturas de SNMP para crear CMA desde CIA de NNMi en la consola de HPOM 49
Integración con la interfaz de reenvío de capturas de NNMi para asignar el origen de la captura de SNMP 50
Configuración del interceptor de captura de SNMP para derivar el nombre de nodo

de origen ... 50
Configuración de la supresión de tormentas de mensajes ... 51
Configuración de la detección y supresión de tormentas de mensajes ... 53
Comprobación de la frecuencia de mensajes ... 57
Configuración del servidor de copia de seguridad ... 57
Configuración del componente RTMA ... 60
Comprobación de la licencia para el proceso Perfd ... 60
Modificación de la configuración ... 60
Monitorización de HPE Operations Agent en las conexiones IPv6 ... 63
Restricción de acceso ... 64
Configuración del usuario del agente ... 65
Requisitos para utilizar un usuario no predeterminado ... 66
Limitaciones en el uso de un usuario no predeterminado ... 67
Configurar el usuario de Agent después de la instalación ... 68
Cambio del usuario predeterminado en Windows ... 68
Método alternativo: Use el comando ovswitchuser ... 68
Cambio del usuario predeterminado en UNIX/Linux ... 71
Usar un archivo de perfil ... 72
Método alternativo: Use el comando ovswitchuser ... 73
Cambio del usuario predeterminado para los comandos ... 75
Configuración del componente de seguridad para la clave asimétrica ... 76
Configuración del componente de seguridad para la clave simétrica ... 79
Configuración del componente de seguridad con el algoritmo hash ... 81
Configuración de HPE Operations Agent para que cumpla los estándares FIPS ... 83
Requisitos para que HPE Operations Agent cumpla los estándares FIPS ... 84
Habilitación del cumplimiento de FIPS con FIPS_tool ... 84
Opciones de configuración en el modo FIPS ... 85
Verificación de si HPE Operations Agent se está ejecutando en el modo FIPS ... 86
Solución de problemas ... 87
Monitorización de los registros de eventos de Windows ... 88
Supervisión de registros de eventos de aplicaciones y servicios desde HPOM para Windows ... 90
Supervisión de registros de eventos de aplicaciones y servicios desde HPOM en UNIX/Linux 9.xx ... 91
Capítulo 2: Asesor del componente RTMA ... 93
Alarmas y síntomas ... 93
Funcionamiento de la secuencias de comandos de asesor ... 94
Uso del asesor ... 94
Ejecución de la secuencia de asesor en múltiples sistemas ... 95
Sintaxis del asesor ... 96
Convenciones de sintaxis ... 96
Comentarios ... 96
Condiciones ... 97
Constantes ... 97
Expresiones ... 98
Nombres de métrica en sintaxis del asesor ... 98

Printlist 100
Variables 100
Instrucción ALARM 101
Instrucción ALERT 102
Instrucción ALIAS 102
Instrucción ASSIGNMENT 102
Instrucción COMPOUND 103
Instrucción EXEC 103
Instrucción IF 103
Instrucción LOOP 104
Instrucción PRINT 105
Instrucción SYMPTOM 105
Capítulo 3: Alarmas de rendimiento 107
Procesamiento de alarmas 107
Generador de alarmas 108
Envío de capturas SNMP a Network Node Manager 110
Envío de mensajes a HPOM 110
Ejecución de acciones locales 111
Errores de procesamiento de alarmas 112
Análisis de datos históricos de alarmas 113
Ejemplos de información de alarma en datos históricos 113
Componentes de definiciones de alarma 114
Referencia de sintaxis de alarma 115
Sintaxis de alarma 115
Convenciones de sintaxis 116
Elementos comunes 116
Nombres de métrica 116
Mensajes 118
Instrucción ALARM 118
Sintaxis 119
Modo en que es utilizado 121
Ejemplos 121
Instrucción ALERT 123
Sintaxis 123
Modo en que es utilizado 124
Ejemplo 124
Instrucción EXEC 124
Sintaxis 125
Modo en que es utilizado 126
Ejemplos 126
Instrucción PRINT 126
Instrucción IF 127
Sintaxis 127
Modo en que es utilizado 127
Ejemplo 127
Instrucción LOOP 128

Sintaxis 129
Modo en que es utilizado 129
Ejemplo 129
Instrucción INCLUDE 130
Sintaxis 130
Modo en que es utilizado 130
Ejemplo 130
Instrucción USE 130
Sintaxis 131
Modo en que es utilizado 131
Instrucción VAR 133
Sintaxis 133
Modo en que es utilizado 133
Ejemplos 134
Instrucción ALIAS 134
Sintaxis 134
Modo en que es utilizado 134
Ejemplos 135
Instrucción SYMPTOM 135
Ejemplos de definiciones de alarma 135
Ejemplo de un problema de CPU 135
Ejemplo de uso de Swap 136
Ejemplo de alarmas basadas en horas 136
Ejemplo de alarmas de instancias de disco 137
Personalización de definiciones de alarma 137
Capítulo 4: HPE Operations Agent en un entorno seguro 139
Directivas 139
Modo de comunicación HTTPS 140
Ventajas de la comunicación HTTPS 140
Agente de comunicación 142
Escenarios de cortafuegos 143
Componentes de seguridad basados en HTTPS 144
Certificados 147
Servidor de certificados de HPE Operations Agent 147
Autoridad de certificados 148
Cliente de certificados 148
Implementación y actualización de certificados raíz 149
Parte III: Uso del componente HPE Operations Agent Performance Collection 150
Capítulo 5: Administración de la recopilación de datos 151
Uso del almacén de datos de métricas 151
Recopilación de datos 153
Verificación del estado del proceso oacore 153
Inicio del proceso oacore 154

Verificación del registro de datos ... 155
Control del espacio en disco usado por los archivos de base de datos ... 156
Control del espacio en disco usado por los archivos de bases de datos que almacenan las clases de métricas de rendimiento predeterminadas ... 156
Control del espacio en disco usado por los archivos de bases de datos que almacenan los datos personalizados ... 158
Detención y reinicio de la recopilación de datos ... 159
Detención de la recopilación de datos ... 159
Reinicio de la recopilación de datos ... 160
Horario de verano ... 161
Cambio manual de la hora del sistema ... 161
Uso del archivo parm ... 161
Instalación de HPE Operations Agent 12.xx ... 161
Actualización a HPE Operations Agent 12.xx ... 162
Modificación del archivo parm ... 163
Parámetros del archivo parm ... 165
Descripciones de los parámetros ... 170
ID ... 170
Log ... 170
Umbrales ... 172
Procthreshold ... 173
appthreshold ... 175
diskthreshold ... 175
bynetifthreshold ... 176
fsthreshold ... 176
lvthreshold ... 176
bycputhreshold ... 176
subprocinterval ... 176
gapapp ... 177
fstypes ... 178
wait ... 179
Size ... 179
javaarg ... 180
Flush ... 180
project_app ... 181
proclist ... 181
appproc ... 181
proccmd ... 181
ignore_mt ... 182
cachemem ... 187
Parámetros de definiciones de aplicación ... 187
Nombre de aplicación ... 188
File ... 189
argv1 ... 191
cmd ... 192
User ... 192

Group ... 193
Or ... 193
Priority ... 194
Ejemplos de definición de aplicación ... 194
Configuración de intervalos de registro de datos ... 195
Configuración de recopilación de datos para marcos ... 196
Tarea 1: Configurar el acceso SSH sin contraseña ... 196
Configurar el acceso SSH sin contraseña ... 196
Comprobación del acceso SSH sin contraseña ... 197
Tarea 2: Habilitar la monitorización de la utilización de los marcos en el sistema HMC 197
Tarea 3: Configurar HPE Operations Agent ... 198
Habilitación de las métricas de llamada del sistema de nivel de proceso y global para GlancePlus en Linux ... 198
Configuración de la recopilación de métricas mediante init_ftrace.sh ... 199
Configuración de la recopilación de métricas mediante pasos manuales ... 200
Normalización de métricas de CPU en sistemas habilitados para hipersubprocesamientos o subprocesamientos múltiples simultáneos ... 203
Registro de métricas calculadas con la normalización basada en núcleo ... 203
Capítulo 6: Uso del programa utility ... 206
Ejecución del programa utility ... 206
Informe de exploración del programa utility ... 209
Comandos del programa utility ... 210
analyze ... 210
checkdef ... 211
detail ... 212
help ... 212
filename ... 213
parmfile ... 213
scan ... 214
Capítulo 7: Uso del programa extract ... 215
Ejecución del programa Extract con la interfaz de línea de comandos ... 215
Uso de la función de exportación ... 218
Cómo exportar datos ... 218
Clases de métricas de rendimiento personalizadas ... 219
Archivos de salida ... 220
Sintaxis del archivo de plantilla de exportación ... 223
Parámetros ... 224
Salida de datos exportados ... 226
Producción de un archivo de salida personalizado ... 227
Notas sobre el formato ASCII ... 228
Comandos del programa extract ... 228
Capítulo 8: Uso del programa cpsh ... 235
Uso del modo interactivo ... 235
Ver métrica en tiempo real ... 236
Modificar una clase de métrica ... 236

Ver toda la métrica disponible ... 237
Organizar una clase de métrica ... 237
Ver Ayuda de métrica ... 237
Ver datos de métrica resumidos ... 238
Habilitación de las condiciones de umbral y de filtro ... 238
Capítulo 9: Creación de recopilaciones de contadores de rendimiento en Windows ... 240
Creación de una recopilación de contadores de rendimiento ... 240
Administración de una recopilación de contadores de rendimiento ... 240
Sugerencias para el uso de Extended Collection Builder and Manager ... 241
Administración de ECBM desde la línea de comandos ... 242
Capítulo 10: Información general del proceso de líneas base ... 245
Configuración de la línea base en el nodo HPE Operations Agent ... 245
¿Cómo solucionar problemas cuando el proceso de líneas base no funciona? ... 249
Capítulo 11: Información general de la resolución de nodos ... 253
Capítulo 12: Registro y rastreo ... 259
Registro ... 259
Configuración de la directiva de registro ... 260
Rastreo ... 260
Identificación de la aplicación ... 262
Establecimiento del tipo de rastreo ... 263
Introducción al archivo de configuración de rastreo ... 264
Sintaxis ... 264
Creación del archivo de configuración ... 265
Habilitación del rastreo y visualización de mensajes de rastreo con las herramientas de la línea de comandos ... 265
Habilitación del rastreo y visualización de mensajes de rastreo con la GUI de rastreo ... 267
Habilitación del mecanismo de rastreo ... 267
Visualización de mensajes de rastreo ... 267
Uso de la vista de la lista de rastreos ... 270
Uso de la vista de árbol de procedimientos ... 271
Filtrado de rastreos ... 271
Uso de la GUI de rastreo ... 273
Parte IV: Registro de datos personalizados ... 279
Capítulo 13: Uso de la interfaz de programación de aplicaciones Perl para enviar datos personalizados ... 280
Envío de datos personalizados ... 282
Envío de datos personalizados a Almacén de datos de métricas ... 283
Caso de uso de envío de datos al almacén de datos mediante API ... 285
Capítulo 14: Introducción general de Integración de Orígenes de Datos (DSI) ... 289
Funcionamiento de DSI ... 290
Capítulo 15: Creación de un modelo para registrar métricas de DSI en el Almacén de datos de métricas ... 293
Creación de un archivo de especificación de clase ... 293

Sintaxis de especificación de clase 294
Descripción de clase 295
Sintaxis 295
CLASS 295
ETIQUETA 296
Registros por hora 296
Configuración predeterminada 297
Especificación de clase de ejemplo 298
Descripciones de métrica 300
ETIQUETA 301
Método de resumen 301
Compilación del archivo de especificación de clase con el compilador DSI 302
Compilador sdlcomp 303
Cambio de especificación de clase 303
Capítulo 16: Registro de métricas DSI en el Almacén de datos de métricas 304
Sintaxis 304
Proceso de registro dsilog 305
Procesamiento de datos de dsilog 305
Administrar datos con sdlutil 306
Sintaxis 306
Capítulo 17: Uso de los datos DSI registros en el Almacén de datos de métricas 307
Definición de alarmas en métrica DSI 307
Procesamiento de alarmas 308
Exportación de datos DSI 308
Ejemplo de uso del programa extract para exportar datos del archivo de registro DSI 308
Ver datos en Performance Manager 309
Capítulo 18: Ejemplos de integración de orígenes de datos (DSI) 310
Escritura de una secuencia de comandos dsilog 310
Ejemplo 1 - Secuencia de comandos dsilog problemática 310
Ejemplo 2 - Secuencia de comandos dsilog recomendada 310
Registro de datos vmstat 311
Creación de un archivo de especificación de clase 311
Compilación de un archivo de especificación de clase 312
Inicio del proceso de registro dsilog 313
Acceso a los datos 313
Registro del número de usuarios del sistema 313
Capítulo 19: Uso de la transmisión de métricas 316
Transmisión de métricas 316
Registro 323
Envío de datos 325
Caracteres especiales en el registro y el envío de datos 328
Configuración del intervalo de publicación 329
Utilización de recursos y escalabilidad 330

Parte V: Seguimiento de transacciones 331
Capítulo 20: Descripción del seguimiento de transacciones 332
Mejora de la administración del rendimiento 332
Ventajas del seguimiento de transacciones 332
Vista del cliente de los tiempos de las transacciones 333
Datos de transacción 333
Objetivos de nivel de servicio 334
Un escenario: Procesamiento de pedidos en tiempo real 334
Requisitos del procesamiento de pedidos en tiempo real 335
Preparación de la aplicación de procesamiento de pedidos 335
Monitorización de datos de transacciones 336
Directrices para usar ARM 337
Capítulo 21: Funcionamiento del seguimiento de transacciones 338
Compatibilidad con ARM 2.0 338
Compatibilidad con llamadas a la API de ARM 339
Llamada arm_complete_transaction 340
Ejemplo de aplicaciones instrumentadas para ARM 340
Especificación de nombres de aplicaciones y transacciones 341
Demonio de seguimiento de transacciones (ttd) 342
Devolución del estado una llamada a la API de ARM 342
Demonio de interfaz de medición (midaemon) 344
Archivo de configuración de transacciones (ttd.conf) 344
Adición de nuevasaplicaciones 345
Adición de nuevas transacciones 345
Cambio de los valores de range o SLO 345
Palabras clave de archivos de configuración 346
tran 346
range 347
slo 347
Formato del archivo de configuración 348
Ejemplos de archivos de configuración 349
Consideraciones de sobrecarga para usar ARM 351
Directrices 351
Sobrecarga de E/S de disco 352
Sobrecargade la CPU 352
Sobrecargade la memoria 352
Capítulo 22: Introducción a las transacciones 354
Antes de comenzar 354
Configuración del seguimiento de transacciones 354
Definición de los objetivos de nivel de servicio 355
Modificación del archivo parm 356
Recopilación de datos de transacciones 356
Control de errores 357
Límites en transacciones exclusivas 357
Personalización del archivo de configuración (opcional) 358

Monitorización de datos de rendimiento ... 359
Alarmas ... 360
Capítulo 23: Mensajes de seguimiento de transacciones ... 362
Capítulo 24: Métricas de transacciones ... 363
Capítulo 25: Ejemplos de seguimiento de transacciones ... 364
Pseudocódigo para el procesamiento de pedidos en tiempo real ... 364
Ejemplos de archivos de configuración ... 368
Ejemplo 1 (ejemplo de pseudocódigo de procesamiento de pedidos) ... 368
Ejemplo 2 ... 368
Ejemplo 3 ... 369
Ejemplo 4 ... 369
Capítulo 26: Funciones avanzadas ... 370
Cómo se usan los tipos de datos ... 370
Métricas definidas por el usuario ... 371
Capítulo 27: Bibliotecas de transacciones ... 374
Biblioteca ARM (libarm) ... 374
Ejemplos de opción de compilador C por plataforma ... 381
Biblioteca ARM NOP ... 383
Uso de los contenedores Java ... 383
Ejemplos ... 383
Configurar una aplicación (arm_init) ... 383
Sintaxis: ... 384
Configurar una transacción (arm_getid) ... 384
Configurar una transacción con UDM ... 384
Agregar la métrica ... 384
Configurar los datos de métrica ... 385
Configurar una transacción sin UDM ... 385
Configurar una instancia de transacción ... 386
Iniciar una instancia de transacción (arm_start) ... 386
Iniciar la instancia de transacción con correladores ... 386
Solicitar un correlador ... 386
Pasar el correlador primario ... 386
Solicitar y pasar el correlador primario ... 387
Recuperar la información del correlador ... 387
Iniciar la instancia de transacción sin correladores ... 387
Actualizar los datos de instancia de transacción ... 387
Actualizar los datos de instancia de transacción con UDM ... 388
Actualizar los datos de instancia de transacción sin UDM ... 388
Proporcionar un búfer privado de aplicación opaco y grande ... 388
Detener la instancia de transacción (arm_stop) ... 389
Detener la instancia de transacción con actualización de métrica ... 389
Detener la instancia de transacción sin actualización de métrica ... 389
Usar la transacción completada ... 390
Usar transacción completada con UDM: ... 390
Usar transacción completada sin UDM: ... 390

Documentación adicional ... 391

Parte VI: Solución de problemas ... 392

Componente Operations Monitoring ... 392

Componente Performance Collection ... 397

RTMA ... 399

GlancePlus ... 399

hpsensor ... 400

Otro ... 400

Parte VII: Actualización a HPE Operations Agent versión 12.xx desde versiones anteriores ... 403

Comparación de HPE Operations Agent 12.xx con versiones anteriores ... 405

Componente Performance Collection ... 409

Archivo parm ... 410

Programa Utility ... 416

Informe de exploración del programa utility ... 418

Información global inicial del archivo parm ... 419

Definiciones de aplicación iniciales del archivo parm ... 420

notificaciones de cambios globales del archivo parm ... 421

Cambios de aplicación del archivo parm ... 421

Notificaciones de tiempo de inactividad de scope. ... 422

Informes de resúmenes específicos de aplicación ... 422

Informe de resumen de proceso ... 423

Inicio y detención de la exploración ... 424

Resumen general de aplicación ... 424

Resumen de cobertura del recopilador ... 425

Resumen de clase y resumen de contenidos del archivo de registro ... 425

Resumen de espacio libre del archivo de registro ... 426

Extract ... 427

Métrica ... 430

Interceptor de capturas de SNMP ... 434

Integración de orígenes de datos ... 436

Preguntas más frecuentes ... 443

Enviar información de la documentación ... 446

Parte I: Introducción

HPE Operations Agent permite monitorizar un sistema recopilando métricas que indican el estado, rendimiento, disponibilidad y utilización de recursos de los elementos fundamentales del sistema.

Con su recopilador de datos integrado, **oacore**, el HPE Operations Agent recopila continuamente los datos de estado y de rendimiento del sistema y almacena los datos recopilados en el Almacén de datos de métricas.

Si se usa HPE Operations Agent en combinación con HPOM, Smart Plug-ins (SPI) y HPE Operations Manager i (HPE OMi), podrá agregarse la funcionalidad de monitorización de las aplicaciones comerciales, la infraestructura (recursos del sistema) así como las cargas de trabajo de la aplicación que están siendo ejecutadas en los sistemas monitorizados.

Las funciones siguientes mejoran la recopilación de datos y las capacidades de monitorización de HPE Operations Agent:

Función	Descripción
parmfile	El mecanismo de recopilación de datos del recopilador de datos de **oacore** está controlado por la configuración del archivo **parm**. El archivo parm en función de las clases definidas en el archivo **parm**, el recopilador de datos **oacore** recopila un gran conjunto de datos que representa una vista amplia del estado y del rendimiento del sistema.
Utility y Extract	Puede usar herramientas como Utility y Extract para ver información específica almacenada en el Almacén de datos de métricas.
Registro de datos personalizados	Puede usar API o DSI de registro de datos personalizados para registrar datos personalizados en el Almacén de datos de métricas.
Creación de líneas base	Puede usar el proceso de creación de líneas base para calcular y proporcionar valores de referencia para analizar tendencias de rendimiento y establecer de manera dinámica valores de umbral óptimos.
Acceso a métricas en	El componente Real Time Metrics Access (RTMA)

tiempo real	proporciona acceso en tiempo real a métricas de rendimiento del sistema.
archivo alarmdef	Se pueden definir las alarmas en el archivo **alarmdef**. A medida que los datos registrados por **oacore** u otros recopiladores, se compara con las definiciones de alarma en el archivo `alarmdef`. Cuando las métricas monitorizadas cumplen o exceden las condiciones definidas, se desencadenará una alerta o acción.
Seguimiento de transacciones	Puede rastrear las transacciones cuando procesan por las aplicaciones. El rastreo de transacciones proporciona una vista de cliente del tiempo transcurrido desde el comienzo al final de una transacción, le ayuda a gestionar los acuerdos de nivel de servicio (SLA) y a generar alarmas cuando los objetivos de nivel de servicio (SLO) exceden las condiciones definidas en el archivo **alarmdef**.

Convenciones usadas en este documento

En este documento se usan las siguientes convenciones:

Convención	Descripción
`<OvInstallDir>` El directorio de instalación de HPE Operations Agent.	**<OvInstallDir>** se usa en este documento para indicar la siguiente ubicación: • *En Windows*:`%ovinstalldir%` • *En HP-UX/Linux/Solaris*: `/opt/OV/` • *En AIX*: `/usr/lpp/OV/`
`<OvDataDir>` El directorio de archivos de datos de tiempo de ejecución y configuración de HPE Operations Agent.	**<OvDataDir>** se usa en este documento para indicar la siguiente ubicación: • *En Windows*: `%ovdatadir%` • *En HP-UX/Linux/Solaris*: `/var/opt/OV/`

	• *En AIX*: `/var/opt/OV/`
`<OvInstallBinDir>` El directorio bin contiene todos los archivos binarios (ejecutables) de HPE Operations Agent.	**<OvInstallBinDir>** se usa en este documento para indicar la siguiente ubicación: • *En Windows x64*: `%ovinstalldir%\bin\win64\` • *En Windows x32*: `%ovinstalldir%\bin\win32\` • *En Windows x86*: `%ovinstalldir%\bin\` • *En HP-UX/Linux/Solaris*: `/opt/OV/bin/` • *En AIX*: `/usr/lpp/OV/bin/`

Parte II: Configuration

Cuando se instala HPE Operations Agent en un entorno de gestión basado en HPOM, puede monitorizar y gestionar los sistemas y aplicaciones desplegados en su entorno de red desde una consola central. Puede usar distintos componentes del componente Operations Monitoring después de desplegar las directivas HPOM en el nodo. Las variables de configuración le permiten configurar el comportamiento predeterminado del HPE Operations Agent. El comando `ovconfchg` permite asignar los valores deseados a estas variables. Puede configurar el agente de monitorización, el interceptor de capturas de SNMP, la supresión de tormenta de mensajes, el servidor de copia de seguridad, el componente RTMA y los componentes de seguridad.

Capítulo 1: Cómo trabajar con HPE 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 Componente Operations Monitoring mediante la implementación de directivas HPOM en el nodo. Por ejemplo, si se despliega una directiva de umbral de medida, el componente del agente de monitorización comienza su cometido. 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 Monitor Agent

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:* Consulte la sección *Event Policy Editors* de la *ayuda en línea de HPOM para Windows*.
- *En HPOM en UNIX/Linux:* Consulte la sección *Implementing Message Policies* de la guía *HPOM for UNIX 9.10 Concepts Guide*.

Configuración del agente de para monitorizar objetos MIB

Después de implementar las directivas de umbral de medicións (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 una sesión en el nodo con privilegios raíz o administrativos.
2. Vaya al símbolo del sistema (shell).
3. Vaya al directorio siguiente:

   En Windows:

   `%ovinstalldir%bin`

   En HP-UX/Linux/Solaris:

   `/opt/OV/bin`

   En 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_comunidad>*

  En este ejemplo, *cadena_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. HPE 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”`

  HPE 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, HPE Operations Agent ejecuta la misma operación con la cadena de comunidad C2, y así sucesivamente.

  Nota: Si HPE 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.

Persistencia de un objeto monitorizado

HPE Operations Agent se puede configurar para que almacene periódicamente los valores de los objetos monitorizados y variables de sesión. El almacenamiento de los valores de los objetos monitorizados y variables de sesión garantiza que los valores se conservar y están disponibles para su uso en caso de una interrupción o error.

La variable OPC_MON_SAVE_STATE permite configurar el agente para que conserve los valores de los objetos monitorizados y variables de sesión.

Para asegurarse de que el agente está configurado para almacenar periódicamente valores de objetos monitorizados y variables de sesión, siga estos pasos:

1. Inicie sesión en el nodo como administrador o usuario raíz.
2. Ejecute el comando siguiente:

   ```
   ovconfchg -ns eaagt -set OPC_MON_SAVE_STATE TRUE
   ```

   El agente empieza a conservar de los valores de los objetos monitorizados y variables de sesión.

La instalación de HPE Operations Agent 12.01 afecta a la variable OPC_MON_SAVE_STATE de la siguiente manera:

- Si no se ha asignado ningún valor a la variable antes de instalar HPE Operations Agent 12.01, la variable OPC_MON_SAVE_STATE asume el valor FALSE.
- Si ha usado el comando ovconfchg para asignar un valor (TRUE o FALSE) a la variable antes de instalar HPE Operations Agent 12.01, el valor configurado permanecerá en vigor aún después de que haya finalizado el proceso de instalación.

Por ejemplo, si ha usado el comando `ovconfchg -ns eaagt -set OPC_MON_SAVE_STATE TRUE` antes de instalar la versión 12.01, se conserva el mismo valor (TRUE) tras la instalación de HPE Operations Agent 12.01.

Mejora de los parámetros de seguridad para ejecutar SNMPv3 GET

En un nodo de HPE Operations Agent, el componente del agente de monitorización (**opcmona**) monitoriza los objetos MIB después de que la directiva de umbral de medición se despliegue en el servidor de gestión HPOM.

Para consultar los datos MIB, **opcmona** ejecuta **SNMPv3 GET** en los ID de objeto (OIDs) especificados. Se requieren parámetros de seguridad adicionales para habilitar **opcmona** para que ejecute **SNMPv3Get**. En la tabla siguiente se muestran estos parámetros:

Parámetros	Descripción
SNMPV3_ENGINEID	Especifica el ID de motor de SNMP que se usa para identificar de manera exclusiva las entidades de SNMP.
SNMPV3_USER	Especifica el nombre de usuario de SNMPv3 creado por el administrador.
SNMPV3_AUTHTYPE	Especifica los protocolos usados para cifrar la contraseña. Puede usar los protocolos Message Digest Algorithm 5 (MD5) o Secure Hash Algorithm (SHA) para cifrar la contraseña.
SNMPV3_ AUTHPASSPHRASE	Especifica el cifrado de contraseña mediante la utilidad **opcpwcrpt**. **Nota:** Para obtener más información, consulte "Mejora de los parámetros de seguridad para ejecutar SNMPv3 GET" arriba.
SNMPV3_ENCRYPTTYPE	Especifica los protocolos usados para cifrar la unidad de datos de protocolo (PDU). Puede usar los protocolos Data Encryption Standard (DES) o Advanced Encryption Standard (AES) para cifrar la PDU.
SNMPV3_	Especifica la clave usada por DES y AES para

ENCRYPTPASSPHRASE	cifrar el PDU. **Nota:** La clave de cifrado creada por el administrador se cifra con la utilidad **opcpwcrpt**. Para obtener más información, consulte "Mejora de los parámetros de seguridad para ejecutar SNMPv3 GET" en la página precedente.

Habilitación de **opcmona** para ejecutar **SNMPv3Get**

Puede habilitar **opcmona** para ejecutar **SNMPv3Get** en uno de los escenarios siguientes:

- Cuando opctrapi se configura usando la directiva de información de nodos
- Cuando opctrapi no se configura usando la directiva de información de nodos

Cuando opctrapi se configura usando la directiva de información de nodos

El ID de motor de una entidad SNMP está disponible en el nodo si **opctrapi** se configura usando la directiva de información de nodos. Para obtener más información, consulte Configuración de opctrapi mediante la directiva informativa de nodo. Agregue el ID de motor de la directiva de umbral de medición para habilitar **opcmona** para que ejecute **SNMPv3Get**.

Nota: Puede ejecutar el comando `ovconfget eaagt` para recuperar el ID de motor. Obtiene la salida siguiente:

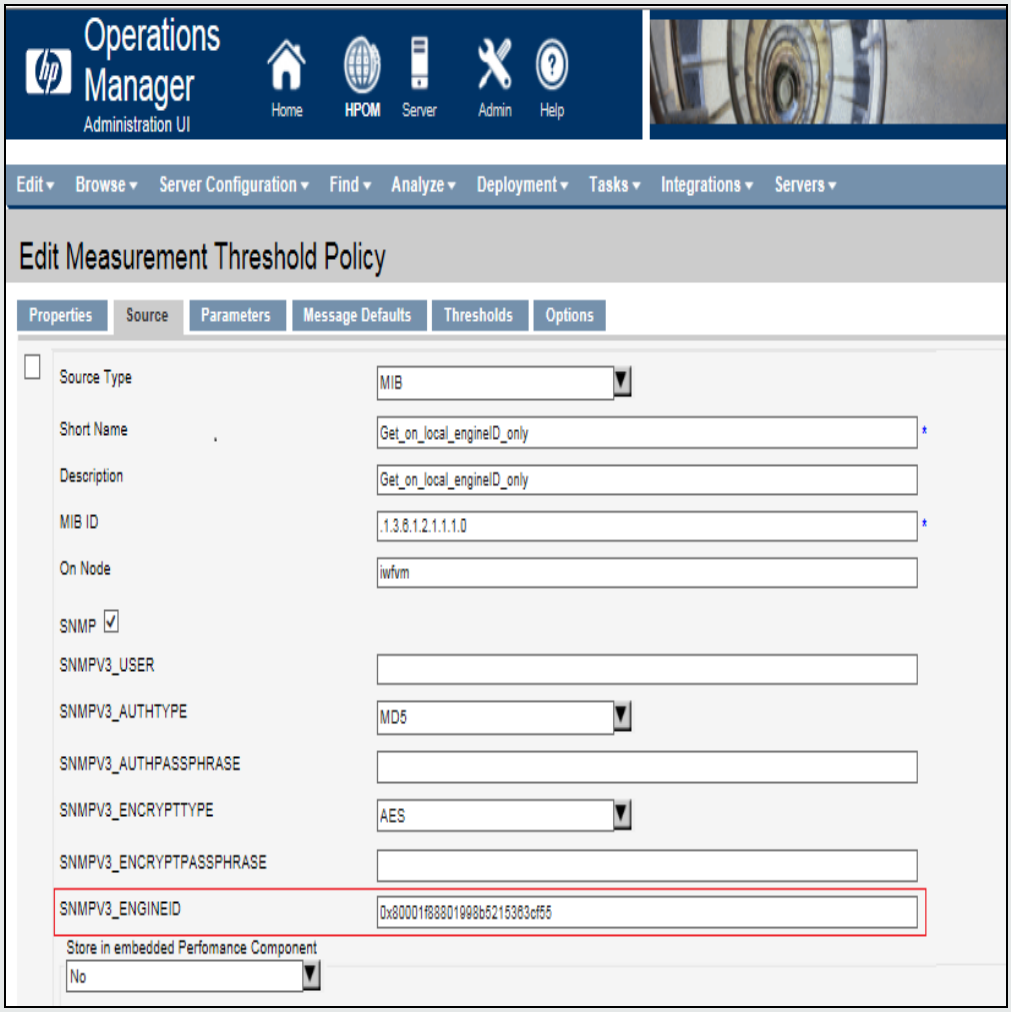
```
OPC_INSTALLATION_TIME=Wed Jan 7 22:42:20 IST 2015
OPC_NODENAME=iwfvm05835.hpswlabs.hp.com
SNMP_V3_
USERSDATA=demosnmpv3user:MD5:7*=*C61Ntd=@*Eb#@*E0##D:DES:7*=*C61Ntd
=@*Eb#@*E0##D:0x80001f88801998b5215363cf55
```

En este ejemplo, el ID de motor es 0x80001f88801998b5215363cf55.

Por ejemplo

Supongamos que el ID de motor que ha usado para configurar **opctrapi** es 0x80001f88801998b5215363cf55.

Agregue el ID de motor de la directiva de umbral de medición como se muestra:



Después de agregar el ID de motor de una entidad SNMP a la directiva de umbral de medición, **opcmona** puede ejecutar **SNMPv3Get** en los OID presentes en esa entidad SNMP.

Cuando opctrapi no se configura usando la directiva de información de nodos

Si **opctrapi** no se configura usando la directiva de información de nodos, debe agregar lo siguiente a la directiva de umbral de medición:

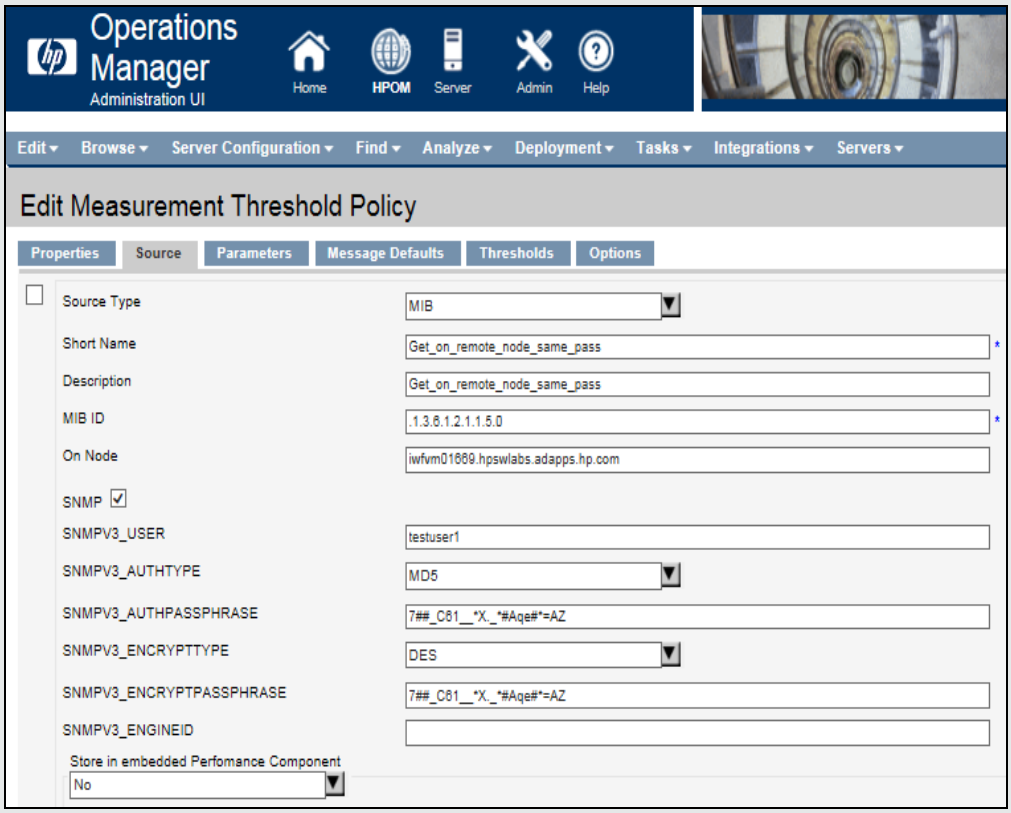
SNMPV3_USER `<nombre de usuario>`
SNMPV3_AUTHTYPE `<método_autenticación>`
SNMPV3_AUTHPASSPHRASE `<contraseña_autenticación>`
SNMPV3_ENCRYPTTYPE `<método_cifrado>`
SNMPV3_ENCRYPTPASSPHRASE `<clave_cifrado>`

Nota:

SNMPV3_AUTHPASSPHRASE y SNMPV3_ENCRYPTPASSPHRASE se cifran automáticamente.

Si las contraseñas de cifrado y de autenticación son las mismas, puede dejar el valor de la variable SNMPV3_ENCRYPTPASSPHRASE en blanco.

Por ejemplo:



Después de agregar los parámetros de una entidad SNMP a la directiva de umbral de medición, **opcmona** puede ejecutar SNMPv3Get en los OID presentes en esa entidad SNMP.

Uso de la API SourceEX para agregar parámetros de seguridad a la directiva

Se introduce una nueva interfaz, **SourceEx_SNMPV3**, para admitir varios parámetros de seguridad usados para monitorizar MIB de SNMPv3. La API SourceEx_SNMPV3 le permite usar los scripts Perl para agregar parámetros de seguridad a la directiva de umbral de medición.

Nota: No puede usar las API SourceEx_SNMPV3 en vbscript de la directiva de umbral de medición.

Use la sintaxis siguiente para agregar parámetros de seguridad en la directiva de umbral de medición en los escenarios siguientes:

1. Cuando **opctrapi** se configura usando la directiva de información de nodos:

```
Policy-> SourceEx_SNMPV3 ("SNMP\\<Id_objeto>[\\<nombre_host>]
","<ID_motor>");
```

Nota: Cuando se especifica el ID de motor, **opcmona** extraerá otros parámetros de seguridad de la directiva de información de nodos desplegada por **opctrapi** para recibir las capturas de SNMPv3. Para obtener más información, consulte Configuración de opctrapi mediante la directiva informativa de nodo.

Por ejemplo:

```
ADVMONITOR "TestMIBviaPerl_V3"
DESCRIPTION "<initial policy version>"
SCRIPTTYPE "Perl"
INSTANCEMODE  SAME
MAXTHRESHOLD
SEPARATORS "               "
SEVERITY Unknown
EXTERNAL "SNMPPolicy"
DESCRIPTION ""
MSGCONDITIONS
DESCRIPTION "New threshold level"
CONDITION_ID "c5e0a2e4-a401-4186-876f-8b20b1ffcd8b"
CONDITION
THRESHOLD
SCRIPT "
use warnings;
my $MetricEx = $Policy->SourceEx_SNMPV3
(\"SNMP\\\\\\\\.1.3.6.1.2.1.1.1.0\\\\\\\\1x.18x.28.7x
\",'0x8000000001020304');
```

En este ejemplo:

Parámetro	Valor
<ID_objeto>	\\.1.3.6.1.2.1.1.1.0
<nombre_host>	\\1x.18x.28.7x
<ID_motor>	0x8000000001020304

2. Cuando **opctrapi** no se configura usando la directiva de información de nodos:

```
$Policy-> SourceEx_SNMPV3 ("SNMP\\<ID_objeto>[\\<nombre_host>]
", "<nombre_usuario>","<método_autenticación>","<contraseña_
autenticación>","<método_cifrado>","<clave_cifrado>");
```

Por ejemplo:

```
ADVMONITOR "TestMIBviaPerl_V3"
DESCRIPTION "<initial policy version>"
SCRIPTTYPE "Perl"
INSTANCEMODE  SAME
MAXTHRESHOLD
SEPARATORS "               "
SEVERITY Unknown
EXTERNAL "SNMPPolicy"
DESCRIPTION ""
MSGCONDITIONS
DESCRIPTION "New threshold level"
CONDITION_ID "c5e0a2e4-a401-4186-876f-8b20b1ffcd8b"
CONDITION
THRESHOLD
SCRIPT "
use warnings;
my $MetricEx = $Policy->SourceEx_SNMPV3
(\"SNMP\\\\\\\\\.1.3.6.1.2.1.1.1.0\\\\\\\\\1x.18x.28.7x\",
\"TestUser\",\"MD5\",'7*=*C61Ntd=@*Eb#@*E0##D',\"DES\",'7*=*C61Ntd
=@*Eb#@*E0##D');
```

En este ejemplo:

Parámetro	Valor
<ID_objeto>	\\.1.3.6.1.2.1.1.1.0

<nombre_host>	\\1x.18x.28.7x
<nombre de usuario>	TestUser
<método_autenticación>	MD5
<contraseña_autenticación>	'7*=*C61Ntd=@*Eb#@*E0##D'
<método_cifrado>	DES
<clave_cifrado>	'7*=*C61Ntd=@*Eb#@*E0##D'

Nota: Asegúrese de que las contraseñas se rodean con comillas simples.

Configuración del Componente Performance Collection de manera remota

Se pueden realizar determinadas tareas de configuración de manera remota en el nodo administrado desde el servidor de administración. En lugar de realizar las tareas de configuración de manera remota para el Componente Performance Collection en cada nodo, se puede utilizar un conjunto especial de directivas y herramientas desde la consola de HP Operations Manager para configurar y trabajar con los nodos de Componente Performance Collection.

Esta función sólo está disponible si instala el paquete de implementación de HPE Operations Agent en los servidores de administración de HP Operations Manager para Windows o HP Operations Manager para UNIX/Linux.

Antes de comenzar

Antes de comenzar a configurar y controlar de manera remota el Componente Performance Collection desde la consola de HP Operations Manager, hay que implementar los archivos de instrumentación del grupo de instrumentación de HPE Operations Agent en aquellos nodos en donde se ejecuta el agente.

Para implementar la instrumentación desde la consola de HP Operations Manager para Windows, siga estos pasos:

Nota: Si se monitorizan nodos de clúster, hay que asegurarse de implementar la instrumentación en todos los nodos que constituyen el clúster y no en el nodo virtual.

1. En el árbol de la consola, haga clic con el botón derecho en el nodo o en el grupo de nodos (donde se está ejecutando el agente) y, a continuación, haga clic en **Todas las tareas > Desplegar instrumental**. Se abrirá el cuadro de diálogo Desplegar instrumental.
2. En el cuadro de diálogo Desplegar instrumental, haga clic en HPE Operations Agent y, a continuación, haga clic en **Aceptar**. La implementación de los archivos de instrumentación necesarios comienza en los nodos.

Para implementar la instrumentación de HP Operations Manager en la consola UNIX/Linux, siga estos pasos:

Nota: Si se monitorizan nodos de clúster, hay que asegurarse de implementar la instrumentación en todos los nodos que constituyen el clúster y no en el nodo virtual.

1. Inicie sesión en la interfaz de usuario de administración.
2. Haga clic en **Deployment > Deploy Configuration**.
3. En la sección Distribution Parameters, seleccione **Instrumentation** y, a continuación, haga clic en **Please Select**. Se abrirá el cuadro emergente **Selector**.
4. En el cuadro emergente **Selector**, seleccione los nodos en donde se está ejecutando el programa de agente.
5. Seleccione la opción **Force Update** para sobrescribir los archivos de instrumentación anteriores.

   Seleccione esta opción en un nodo que se actualizó desde una versión anterior del agente.
6. Haga clic en **Distribute**.

Implementación de la directiva OA-PerfCollComp-opcmsg

La directiva OA-PerfCollComp-opcmsg envía los mensajes de alerta al explorador de mensajes de HPOM cuando el Componente Performance Collection genera las alarmas. La directiva está ubicada en el grupo de directivas **HP Operations Agent > Componente Performance Collection > Interceptor de mensajes**. Antes de implementar otras directivas para el Componente Performance Collection, hay que implementar esta directiva en los nodos.

Nota: Si se monitorizan nodos de clúster, hay que asegurarse de implementar la directiva en todos los nodos que constituyen el clúster y no en el nodo virtual.

Configuración del Componente Performance Collection

El comportamiento del Componente Performance Collection de HPE Operations Agent depende de los ajustes de la configuración especificada en los archivos siguientes:

- Archivos de parámetros de la recopilación (**parm**)
- Archivo de definición de alarmas (**alarmdef**)

Consulte la sección *Componente Performance Collection* de la guía *HPE Operations AgentConcepts Guide* para obtener más información sobre los parámetros de recopilación y los archivos de definición de alarmas.

Configuración del archivo parm

El archivo **parm** define el mecanismo de recopilación de datos del recopilador **oacore**. HPE Operations Agent despliega un archivo **parm** en cada nodo, que está disponible en la siguiente ruta de acceso:

En HP-UX, Solaris, AIX y Linux: **/var/opt/perf**

En Windows: *%ovdatadir%*

La configuración especificada en el archivo **parm** se puede modificar para personalizar el mecanismo de recopilación de datos. Sin embargo, si se administra un gran número de nodos con HPE Operations Agent, puede resultar difícil modificar cada copia del archivo **parm** en cada nodo.

Con la consola de HP Operations Manager, se puede implementar el archivo **parm** modificado de manera central en varios nodos desde el servidor de administración.

En HP Operations Manager para Windows

La consola de HP Operations Manager para Windows ofrece directivas ConfigFile que ayudarán al usuario a implementar los cambios realizados en el archivo **parm** por varios nodos desde el servidor de administración central. Hay varias directivas ConfigFile disponibles para los distintos sistemas operativos de los nodos.

Para modificar el mecanismo de recopilación editando el archivo **parm**, siga estos pasos:

1. Identifique los nodos en donde desea que surta efecto el mecanismo de recopilación modificado.

2. En el árbol de la consola, haga clic en **Gestión de directivas** > **Grupos de directivas** > **HPE Operations Agent** > **Componente Performance Collection** > **Configuración de recopilación**. Las directivas ConfigFile para configurar el archivo **parm** aparecerán en el panel de detalles.
3. Haga doble clic en la directiva ConfigFile para la plataforma en la que desea que surta efecto el mecanismo de recopilación modificado (por ejemplo: archivo **parm** para HP-UX). Se abrirá el cuadro de diálogo **parm** para *<plataforma>*.
4. En la pestaña Datos, modifique la configuración. Consulte la sección *Parámetros del archivo parm* en la *Guía de usuario de HPE Operations Agent* para obtener más información sobre los parámetros de configuración del archivo **parm**.
5. Haga clic en **Guardar y cerrar**. En el panel de detalles, la versión de la directiva aumenta en .1.
6. Implemente la directiva actualizada en los nodos que prefiera.

   Nota: Si se monitorizan nodos de clúster, hay que asegurarse de implementar la directiva en todos los nodos que constituyen el clúster y no en el nodo virtual.

En HP Operations Manager en UNIX/Linux 9.10

En la consola de HP Operations Manager en UNIX/Linux 9.10 se encontrarán directivas ConfigFile que ayudarán al usuario a implementar los cambios realizados en el archivo **parm** por varios nodos desde el servidor de administración central. Hay varias directivas ConfigFile disponibles para los distintos sistemas operativos de los nodos.

Para modificar el mecanismo de recopilación editando el archivo **parm** desde la consola de HP Operations Manager para UNIX 9.10, siga estos pasos:

1. Identifique los nodos en donde desea que surta efecto el mecanismo de recopilación modificado.
2. En la consola, haga clic en **Examinar** > **Todos los grupos de directivas**. En la página se mostrará la lista de todos los grupos de directivas disponibles.
3. Haga clic en **H**. Se mostrará el grupo de directivas de HPE Operations Agent.
4. Haga clic sucesivamente en **HP Operations Agent**, **Componente de recopilación de rendimiento** y, a continuación, en **Configuración de recopilación**. Se mostrará una lista de las directivas ConfigFile disponibles para el archivo **parm**.

5. Haga clic en la directiva ConfigFile para la plataforma en la que desea que surta efecto el mecanismo de recopilación modificado. Se mostrará la página Policy "OA_<*plataforma*>ParmPolicy".
6. Haga clic en  y, a continuación, haga clic en **Edit (Raw Mode)**. Se mostrará la página Edit Config File policy...
7. En la pestaña Contenido, modifique la configuración.

   Consulte la sección *Parámetros del archivo parm* en la *Guía de usuario de HPE Operations Agent* para obtener más información sobre los parámetros de configuración del archivo **parm**.
8. Haga clic en **Guardar**.
9. Implemente la directiva actualizada en los nodos que prefiera.

Nota: Si se monitorizan nodos de clúster, hay que asegurarse de implementar la directiva en todos los nodos que constituyen el clúster y no en el nodo virtual.

Configuración del archivo alarmdef

El archivo de definición de alarmas (**alarmdef**) proporciona al subagente de rendimiento la especificación predeterminada para el proceso de generación de alarmas. HPE Operations Agent despliega un archivo **alarmdef** en cada nodo, que está disponible en la siguiente ruta de acceso:

En HP-UX, Solaris, AIX y Linux: `/var/opt/perf/`

En Windows: `%ovdatadir%`

Se puede modificar la configuración predeterminada del archivo **alarmdef** para personalizar el mecanismo de generación de alarmas. En la consola de HP Operations Manager se puede distribuir centralmente el archivo **alarmdef** modificado en varios nodos.

En HP Operations Manager para Windows

La consola de HP Operations Manager para Windows ofrece directivas ConfigFile que ayudarán al usuario a implementar los cambios realizados en el archivo **alarmdef** por varios nodos desde el servidor de administración central. Hay varias directivas ConfigFile disponibles para los distintos sistemas operativos de los nodos.

Para modificar el mecanismo de recopilación mediante la edición del archivo **alarmdef**, siga estos pasos:

Identifique los nodos en donde desea que surta efecto el mecanismo de recopilación modificado.

1. En el árbol de consola, haga clic en **Gestión de directivas** > **Grupos de directivas** > **HP Operations Agent** > **Componente Performance Collection** > **Definición de alarma**. Las directivas ConfigFile para configurar el archivo **alarmdef** aparecerán en el panel de detalles.
2. Haga doble clic en la directiva ConfigFile para la plataforma en la que desea que surta efecto el mecanismo de recopilación modificado (por ejemplo: archivo Alarmdef para HP-UX). Se abre el cuadro de diálogo <*plataforma*>.
3. En la pestaña Datos, modifique la configuración. Consulte la sección *Parámetros del archivo alarmdef* de la *Guía del usuario de HPE Operations Agent* para obtener más información sobre los parámetros de configuración del archivo **alarmdef**.
4. Haga clic en **Guardar y cerrar**. En el panel de detalles, la versión de la directiva aumenta en .1.
5. Implemente la directiva actualizada en los nodos que prefiera.

Nota: Si se monitorizan nodos de clúster, hay que asegurarse de implementar la directiva en todos los nodos que constituyen el clúster y no en el nodo virtual.

En HP Operations Manager en UNIX/Linux 9.10

En la consola de HP Operations Manager en UNIX/Linux 9.10 se encontrarán directivas ConfigFile que permitirán implementar los cambios realizados en el archivo **alarmdef** por varios nodos desde el servidor de administración central. Hay varias directivas ConfigFile disponibles para los distintos sistemas operativos de los nodos.

Para modificar el mecanismo de recopilación editando el archivo **alarmdef** desde la consola de HP Operations Manager para UNIX 9.10, siga estos pasos:

1. Identifique los nodos en que desea que surta efecto el mecanismo de alerta modificado.
2. En la consola, haga clic en **Examinar** > **Todos los grupos de directivas**. En la página se mostrará la lista de todos los grupos de directivas disponibles.
3. Haga clic en **H**. Se mostrará el grupo de directivas de HP Operations Agent.
4. Haga clic sucesivamente en **HP Operations Agent**, **Componente de recopilación de rendimiento** y, a continuación, en **Definición de alarma**. Se mostrará una lista de las directivas ConfigFile disponibles para el archivo **alarmdef**.

5. Haga clic en la directiva ConfigFile para la plataforma en la que desea que surta efecto el mecanismo de recopilación modificado. Aparecerá la página Policy “OA_<*plataforma*>AlarmdefPolicy”.
6. Haga clic en  y, a continuación, haga clic en **Edit (Raw Mode)**. Se mostrará la página Edit Config File policy...
7. En la pestaña Contenido, modifique la configuración. Consulte la sección *Parámetros del archivo alarmdef* de la *Guía del usuario de HPE Operations Agent* para obtener más información sobre los parámetros de configuración del archivo **alarmdef**.
8. Haga clic en **Guardar**.
9. Implemente la directiva actualizada en los nodos que prefiera.

Nota: Si se monitorizan nodos de clúster, hay que asegurarse de implementar la directiva en todos los nodos que constituyen el clúster y no en el nodo virtual.

Trabajar de manera remota con HPE Operations Agent

La consola de HP Operations Manager se puede utilizar para iniciar, detener, monitorizar y visualizar los detalles de HPE Operations Agent. La consola de HP Operations Manager ofrece distintas herramientas para administrar el funcionamiento de HPE Operations Agent. Estas herramientas se deben iniciar en los nodos en donde se implementa el agente. En la sección siguiente, se muestra el resultado de la ejecución de una herramienta:

HP Operations Manager para Windows

Sección Salida de la herramienta de la ventana Estado de las herramientas

HP Operations Manager en UNIX/Linux

En la ventana Salida de aplicación de la GUI de Java (Interfaz operativa de HP Operations Manager for Unix)

Puede usar las herramientas siguientes en la consola de HP Operations Manager:

Herramienta	Descripción
Start Agent	Permite iniciar HPE Operations Agent en el nodo administrado.
Stop Agent	Permite detener HPE Operations Agent en el nodo administrado.

Restart Agent	Permite reiniciar HPE Operations Agent en el nodo administrado.
View Status	Permite ver el estado del proceso, servicios y demonios de HPE Operations Agent en el nodo administrado.
View Version Information	Permite ver la versión de HPE Operations Agent en el nodo administrado.
Refresh Alarm Service	Actualiza el servicio de alarma del Componente Performance Collection.
Scan Performance Component's Log Files	Examina los archivos de registro usados por el recopilador de ámbito en el nodo.
Check Performance Component's Parameter File Syntax	Permite comprobar la sintaxis del archivo de parámetros en el nodo administrado.
Check Performance Component's Alarmdef File Syntax	Permite comprobar la sintaxis del archivo alarmdef en el nodo administrado.
View status of post policy deploy action	Permite comprobar el estado de implementación de las directivas parm o alarmdef en los nodos. Al iniciar esta herramienta, hay que asegurarse de especificar parm o alarmdef (según corresponda) como parámetro de la herramienta. Al usar HP Operations Manager para Windows, el parámetro de la herramienta se puede definir en el cuadro Parámetro de la ventana Editar parámetros. Al usar HP Operations Manager en UNIX/Linux, hay que abrir la página Edit Tool Status en la herramienta, ir a la pestaña OVO Tool y, a continuación, especificar el parámetro de la herramienta en el cuadro Parameters.
Set Realtime Permanent License	Establece la licencia permanente del HP Ops OS Inst en Realtime Inst LTU.

Set Glance Permanent License	Establece la licencia permanente de Glance Software LTU.
Get License Status	Muestra el estado de LTU en el nodo.

Configuración del interceptor de capturas de SNMP

De manera predeterminada, el interceptor de capturas de SNMP puede recopilar capturas de 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.

Nota: El interceptor de capturas de SNMP `(opctrapi)` no aplica formato a la parte textual de MIB. Por ejemplo, el texto de mensaje muestra MIB como `.1.3.6.1.4`, y no como `.iso.identified-organization.dod.internet.private`.

Se puede modificar el comportamiento predeterminado del interceptor de capturas de SNMP 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 HPE Operations Agent.
- SNMP_TRAP_FORWARD_DEST_LIST: Usa esta propiedad para establecer la dirección de la estación de gestión remota donde desee reenviar todas las capturas de 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 capturas de SNMP para reenviar las capturas de SNMP disponibles en el nodo de HPE Operations Agent a equipos remotos o estaciones de administración.
- SNMP_TRAP_FORWARD_COMMUNITY: Use esta propiedad para especificar la cadena de comunidad de los equipos de origen de las capturas entrantes y el equipo de destino donde desee reenviar las capturas de SNMP. Las cadenas de comunidad de los equipos de origen deben coincidir con las de los equipos de destino.
- SNMP_TRAP_FORWARD_FILTER: Use esta propiedad para filtrar las capturas de SNMP disponibles por sus OID y reenviar solo 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 capturas de SNMP reenviará todas las capturas de 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 capturas de SNMP para que reenvíe capturas.

Nota: 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 capturas de SNMP, 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 capturas de SNMP para reenviar capturas de SNMP a equipos remotos, ejecute el comando siguiente:

     `ovconfchg -ns eaagt set SNMP_TRAP_FORWARD_ENABLE TRUE`

   - Si se habilita el interceptor de capturas de SNMP para que reenvíe capturas de 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>:<port>*`"`

     <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 capturas de SNMP 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 interceptor de capturas de SNMP para capturas de SNMPv3

En un entorno en red, es importante asegurar la comunicación segura entre el dispositivo que envía capturas y el HPE Operations Agent. El protocolo simple de administración de redes versión 3 (SNMPv3) proporciona un acceso seguro a los dispositivos de envío de capturas al autenticar usuarios y cifrar datos de los paquetes enviados por la red.

En las versiones anteriores deHPE Operations Agent, el proceso **opctrapi** estaba configurado para interceptar las capturas de SNMPv1 y SNMPv2. Con la versión HPE Operations Agent 12.01, **opctrapi** también puede interceptar los mensajes *trap* e *inform* de SNMPv3.

Configure las variables siguientes para habilitar **opctrapi** e interceptar capturas de SNMPv3:

Nota: El proceso **opctrapi** intercepta capturas de SNMPv3 solo en el modo NETSNMP.

- SNMP_V3_ENABLE - Esta variable es obligatoria y debe establecerse en el espacio de nombres `eaagt`. De manera predeterminada, esta variable está establecida en *TRUE*.

  Para deshabilitar opctrapi de interceptar capturas de SNMPV3, establezca SNMP_V3_ENABLE en *FALSE*. Si la variable se establece en *FALSE*, **opctrapi** intercepta solo capturas de SNMPv1 y SNMPv2.
- SNMP_V3_USERSDATA - Esta variable es obligatoria y debe establecerse en el espacio de nombres `eaagt`. Use la variable para configurar usuarios.

  SNMP_V3_USERSDATA = `<parámetros_de_usuario_1>;<parámetros_de_ usuario_2>;<parámetros_de_usuario_n>`

  En este caso, `<parámetros_de_usuario>` incluye lo siguiente:

  `<nombre_de_usuario>:<método_de_autenticación>:<contraseña_de_autenticación>:<método_de_cifrado>:<clave_de_cifrado>:<ID_de_motor>`

Parámetro	Descripción
`<nombre de usuario>`	Especifica el nombre de usuario de SNMPv3 creado por el administrador.

`<método_autenticación>`	Especifica los protocolos usados para cifrar la contraseña. Puede usar los protocolos Message Digest Algorithm 5 (MD5) o Secure Hash Algorithm (SHA) para cifrar la contraseña.
`<contraseña_autenticación>`	Especifica el cifrado de contraseña mediante la utilidad **opcpwcrpt**. **Nota:** Para obtener más información, consulte Cifrado de contraseñas con la utilidad opcpwcrpt.
`<método_cifrado>`	Especifica los protocolos usados para cifrar la unidad de datos de protocolo (PDU). Puede usar los protocolos Data Encryption Standard (DES) o Advanced Encryption Standard (AES) para cifrar la PDU.
`<clave_cifrado>`	Especifica la clave usada por DES y AES para cifrar el PDU. **Nota:** La clave de cifrado creada por el administrador se cifra con la utilidad **opcpwcrpt**. Para obtener más información, consulte Cifrado de contraseñas con la utilidad opcpwcrpt.
`<ID_motor>`	Especifica el ID de motor de SNMP que se usa para identificar de manera exclusiva las entidades de SNMP.

Configuración de opctrapi para interceptar capturas de SNMPv3

Puede usar cualquiera de los siguientes métodos para configurar opctrapi para que intercepte capturas de SNMPv3:

- Uso de la directiva de información de nodos
- Uso de las variables XPL

Uso de la directiva de información de nodos

Siga estos pasos:

1. Inicie sesión en el servidor HPOM.
2. Agregue las variables siguientes a la directiva de información de nodos:
   a. Para habilitar SNMPv3 , establezca la variable SNMP_V3_ENABLE en *True*

      `SNMP_V3_ENABLE = TRUE`

   b. Para configurar usuarios, ejecute el comando siguiente:

      `SNMP_V3_USERSDATA = <parámetros para usuario 1>;<parámetros para usuario 2>;<parámetros para usuario n>`
3. Despliegue la directiva de información de nodos en los nodos. `opctrapi` se reinicia automáticamente.

A continuación se muestra un ejemplo de un fragmento de código en la directiva de información de nodos para configurar **opctrapi** para interceptar capturas de SNMPv3:

```
;XPL config
[eaagt]
SNMP_V3_ENABLE=TRUE
SNMP_V3_USERSDATA=<parámetros _de_ususario_1>;<parámetros _de_
ususario_2>;<parámetros _de_ususario_n>
```

Nota: El tipo de directiva de información de nodos proporciona una forma para modificar la información de configuración en un nodo gestionado donde se ejecuta HPE Operations Agent.

Uso de las variables XPL

Siga estos pasos:

1. Inicie sesión en el nodo de HPE Operations Agent.
2. Para habilitar la interceptación de capturas de SNMPv3, ejecute el comando siguiente:

   `ovconfchg -ns eaagt -set SNMP_V3_ENABLE TRUE`
3. Para establecer usuarios, ejecute el comando siguiente:

   `ovconfchg -ns eaagt -set SNMP_V3_USERSDATA` *<parámetros _de_ ususario_1>;<parámetros _de_ususario_2>;<parámetros _de_ususario_n>*

   opctrapi se reinicia automáticamente.

   Por ejemplo:

   `ovconfchg -ns eaagt -set SNMP_V3_ENABLE TRUE`

```
ovconfchg -ns eaagt -set SNMP_V3_USERSDATA
"snmpv3User1:SHA:7*=*C61Ntd=@*Eb#@*E0##D:DES:7*=*C61Ntd=@*E
b#@*E0##D:0x8000000001020304;snmpv3User2:SHA:8*=*D61Ntd=@*E
b#@*E0##D:DES:8*=*D61Ntd=@*Eb#@*E0##D:0x8000000001030404;"
```

En este ejemplo:

Parámetro	Valor:	Valor:
`<nombre de usuario>`	snmpv3User1	snmpv3User2
`<método_autenticación>`	SHA	SHA
`<contraseña_autenticación>`	7*=*C61Ntd=@*Eb#@*E0##D	8*=*D61Ntd=@*Eb#@*E0##D
`<método_cifrado>`	DES	DES
`<clave_cifrado>`	7*=*C61Ntd=@*Eb#@*E0##D	8*=*D61Ntd=@*Eb#@*E0##D
`<ID_motor>`	0x8000000001020304	0x8000000001030404

Nota: Cada parámetro se separa del siguiente por dos puntos (:) y cada usuario se separa del siguiente por un punto y coma (;).

Cifrado de contraseñas con la utilidad **opcpwcrpt**

1. Inicie sesión en el servidor HPOM con privilegios administrativos.
2. Abra el símbolo del sistema.
3. Vaya a la ubicación siguiente:

   En Windows:

   `"%OvBinDir%\OpC\install"`

   En Linux:

   `/opt/OV/bin/OpC/install`
4. Ejecute el comando siguiente para cifrar la contraseña:

   `# opcpwcrpt <su_contraseña>`

La cadena de salida es la contraseña cifrada. Use la contraseña cifrada de manera adecuada en lugar de `<contraseña_autenticación>` o `<clave_cifrado>`.

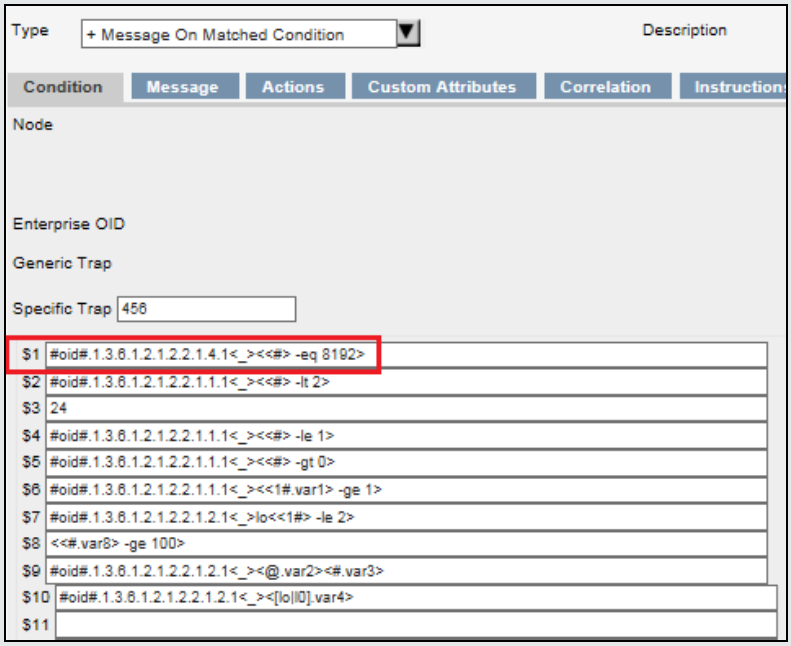
Mejora del interceptor de capturas de SNMP para interceptar capturas basadas en el Id. de objeto del enlace de variables

El interceptor de capturas de SNMP **opctrapi** ha mejorado para interceptar capturas de SNMP basadas en el Id. de objeto de los enlaces de variables (OID) junto con la posición.

Para que **opctrapi** intercepte capturas de SNMP basadas en el Id. de objeto del enlace de variables, puede crear o modificar una directiva para el interceptor de capturas de SNMP y agregar (o modificar) lo siguiente, tal como se muestra en el ejemplo:

```
<keyword><object id><separator><pattern string>
```

Por ejemplo:



En la instancia marcada de la captura de pantalla:

`<keyword>` siempre es “#oid#”

`<object id>` es .1.3.6.1.2.1.2.2.1.4.1

`<separator>` es <_>

`<pattern string>` es <<#> -eq 8192>

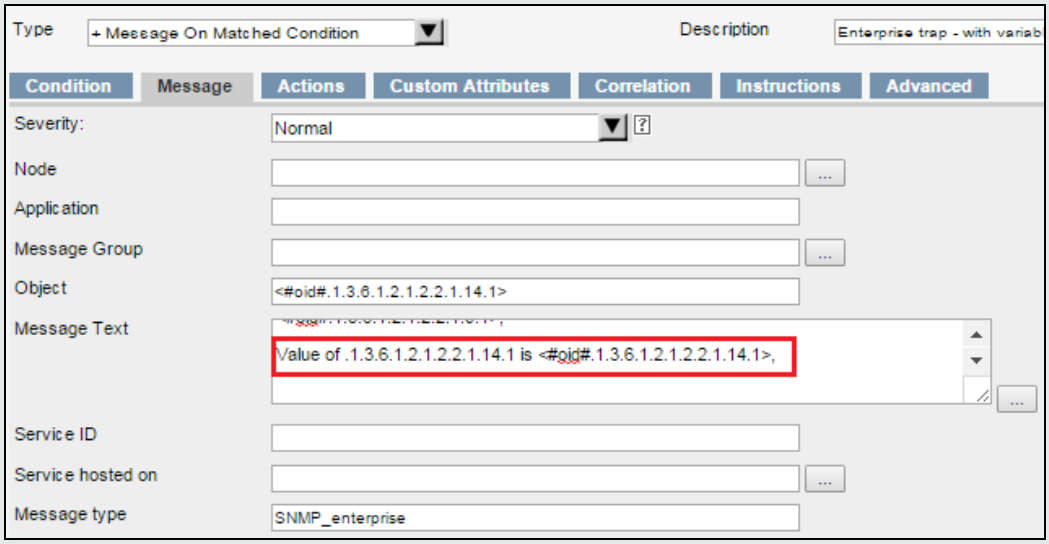
El valor es 8192.

Aunque coincide con el enlace de variables, **opctrapi** comprueba si el campo comienza con la palabra clave **#oid#**. Si el campo comienza con la palabra clave **#oid#**, el OID del enlace de variables de la captura se compara con la cadena de patrón mencionada en el campo y la condición se verifica.

Nota: En la directiva de interceptor de capturas de SNMP, también puede usar OID en:

- Atributos de mensajes (texto de mensaje, grupo de mensajes, nombre de servicio, objeto de mensaje, clave de mensaje, objeto, aplicación)
- Atributos de mensaje personalizados (CMA)

En el ejemplo siguiente se usa OID en el texto del mensaje:



Habilitar opctrapi para usar el registro de NETSNMP

El proceso **opctrapi** registran los mensajes de seguimiento para las capturas SNMPv3 en el archivo `<%ovdatadir%>/tmp/opc/trace`. Estos mensajes no proporcionan suficiente información necesaria para solucionar problemas relacionados con los parámetros de seguridad asociados a las capturas de SNMPv3 entrantes.

Para resolver este problema, el proceso **opctrapi** se ha mejorado para usar el registro de NETSNMP. Debe establecer las siguientes variables de seguimiento en el espacio de nombres `eaagt` para habilitar **opctrapi** con el fin de usar el registro de NETSNMP:

- OPC_TRACE: de manera predeterminada, esta variable está establecida en FALSE. Establezca esta variable en TRUE para habilitar el seguimiento.
- OPC_TRC_PROCS: esta variable enumera los nombre de todos los procesos de los que es necesario realizar un seguimiento. Los nombres de los procesos se deben separar con comas.
- OPC_DBG_PROCS: esta variable enumera los nombre de todos los procesos de los que se ha realizado un seguimiento y que es necesario depurar. Los nombres de los procesos se deben separar con comas.

Por ejemplo:

OPC_TRACE=TRUE

OPC_TRC_PROCS=opctrapi

OPC_DBG_PROCS=opctrapi

Después de definir estas variables, se crea un nuevo archivo de registro `netsnmp.log` en `<%datadir%>/tmp/OpC`. El archivo `netsnmp.log` contiene información completa acerca de errores y capturas entrantes que pueden usarse para solucionar problemas.

Por ejemplo:

Puede comprobar las siguientes condiciones de seguimiento en el archivo `netsnmp.log`:

- Error de usuario desconocido: este error se registra si un usuario de SNMPv3 no está configurado y se recibe una captura de SNMPv3.
- Error de autenticación: este error se registra si el método de autenticación o la contraseña de SNMPv3 son incorrectos.
- Error de cifrado: este error se registra la clave o el método de cifrado de SNMPv3 son incorrectos.

Nota: El archivo `netsnmp.log` no se sustituye a intervalos regulares. Los mensajes de seguimiento se añaden al mismo archivo, incluso si vuelve a habilitar el seguimiento.

Integración de HPE Operations Agent con NNMi

El software HP Network Node Manager i (HP NNMi) es un software de gestión de red que usa agentes SNMP existentes para la detección y estado de elementos de red. La integración con HPE Operations Agent le ayuda a monitorizar las capturas reenviadas por NNMi y ver las capturas de SNMP enriquecidas en la consola de HPOM.

La integración de HPE Operations Agent con HP NNMi está disponible con lo siguiente:

- Interfaz Northbound: la interfaz Northbound de NNMi se usa para reenviar incidentes de NNMi a cualquier aplicación que pueda recibir capturas de SNMPv2c. Los eventos que se envían al interceptor de capturas de SNMP **(opctrapi)** desde NNMi utilizando la interfaz Northbound. Para garantizar la asociación correcta entre el dispositivo que envía la captura y el evento en la aplicación receptora, las reglas de esas capturas deben personalizarlas los enlaces de variables. La integración se realiza mediante la directiva que contiene las reglas de **opctrapi** para interceptar capturas reenviadas por la interfaz Northbound de NNMi. **opctrapi** también establecerá varios parámetros del mensaje de salida basado en las distintas configuraciones, como la gravedad o nombre del nodo, entre otros, definidas por la directiva. Para obtener más información sobre el uso de la interfaz Northbound, consulte *NNMi Deployment Reference, version 9.22.*
- Reenvío de eventos: el mecanismo de reenvío de capturas de SNMP de NNMi enriquece cada captura de SNMP antes de reenviarlas al destino de capturas. Hay dos tipos de mecanismos de reenvío de eventos:
  - Reenvío de captura original: no se agregan enlaces de variables adicionales.
  - Reenvío de captura predeterminado: NNMi agrega enlaces de variables para identificar el objeto de origen original. Este mecanismo lo utiliza **opctrapi** para mejorar las capturas de SNMP.

Para obtener más información sobre el reenvío de capturas predeterminado, consulte *Configuring Trap Forwarding in the NNMi Online Help*.

NNMi reenvía los incidentes mediante el reenvío de eventos de NNMi o una interfaz Northbound y agrega los enlaces de variables al evento original. Estos enlaces de variable contienen información adicional de las capturas. Para procesar y usar esta información con HPE Operations Agent, debe configurar el interceptor de SNMP (**opctrapi**) para lo siguiente:

- Asociar la gravedad que está disponible con la captura de SNMP con el mensaje de HPOM. Esto solo funciona con la interfaz Northbound. Para obtener más información, consulte Integración con la interfaz Northbound de NNMi para asociar la gravedad de los mensajes.
- Cree atributos de mensaje personalizados (CMA) de HPOM desde incidentes de NNMi: esto solo funciona con la interfaz Northbound. Para obtener más información, consulte Integración con la interfaz Northbound para agregar CMA a la consola de HPOM.
- Derivar el nombre de nodo de origen donde se originó la captura de SNMP. Esto solo funciona con el reenvío de capturas predeterminado. Para obtener más información, consulte Integración con la interfaz de reenvío de capturas de NNMi para asignar el origen de la captura de SNMP.

Integración con la interfaz Northbound de NNMi para asociar la gravedad de los mensajes.

HPE Operations Agent usa la directiva que genera la herramienta NNMi `nnmopcexport.ovpl`. Esta herramienta establece la gravedad de las capturas de SNMP. Si la gravedad generada por la herramienta NNMi es normal, crítica, aviso, grave o leve, la gravedad del mensajes aparece como normal, crítica, aviso, grave o leve.

Solo cuando la herramienta indica la gravedad del mensaje generado por la directiva HPOM como **Desconocido**, configure opctrapi para usar el enlace de variable disponible en la captura para derivar el valor de la gravedad. Esta configuración ayuda a asociar la gravedad disponible con la captura de SNMP en la gravedad de mensajes HPOM.

Configuración del interceptor de capturas de SNMP para mejorar la gravedad de los mensajes

Para establecer la gravedad de los mensajes en función del nivel de gravedad disponible en la captura de SNMP, puede configurar el interceptor de capturas de SNMP **(opctrapi)**. Interprete la gravedad del OID específico de la captura de SNMP y después asocie el mismo nivel de gravedad a los mensajes de HPOM. Este paso de configuración ayuda a opctrapi para usar el nivel de gravedad disponible en las capturas de SNMP.

Nota: La configuración es aplicable a las reglas en las que la gravedad de los mensajes establecida en HPOM es desconocido.

Se puede modificar el comportamiento predeterminado de **opctrapi** mediante la configuración de las propiedades siguientes:

- OPC_SNMP_SET_SEVERITY: De manera predeterminada, esta propiedad está establecida en **FALSE**. Al establecer esta propiedad en **TRUE**, puede habilitar el interceptor de capturas de SNMP para interpretar las capturas de SNMP con el OID del enlace de variable específico **(.1.3.6.1.4.1.11.2.17.19.2.2.12**) y establecer la gravedad del mensaje. Si este valor de OID predeterminado no está disponible en la captura de SNMP, la gravedad del mensaje sigue siendo desconocida.
- OPC_SNMP_OVERRIDE_SEVERITY_OID: Puede establecer el nuevo valor de OID. El nuevo valor especificará la gravedad del mensaje.

Para configurar el interceptor SNMP, siga estos pasos:

1. Inicie sesión en el nodo con los privilegios necesarios.
2. En el símbolo del sistema,
   - Para habilitar el interceptor SNMP para que interprete los niveles de gravedad definidos en las capturas de SNMP, ejecute el comando siguiente:

     ```
     ovconfchg -ns eaagt.integration.nnm -set OPC_SNMP_SET_
     SEVERITY TRUE
     ```

   - Para establecer el nuevo valor de OID, ejecute el comando siguiente:

     ```
     ovconfchg -ns eaagt.integration.nnm -set OPC_SNMP_OVERRIDE_
     SEVERITY_OID <OID>
     ```

     En el anterior comando, <*OID*> es el identificador de objeto. Aquí, OID se usa para derivar el nivel de gravedad de las capturas de SNMP.

     De manera predeterminada, el nivel de gravedad procede del OID predeterminado **".1.3.6.1.4.1.11.2.17.19.2.2.12"**. Puede cambiar el OID predeterminado ejecutando el comando siguiente:

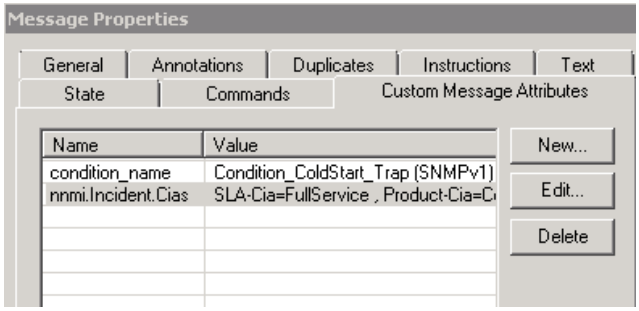
     ```
     ovconfchg -ns eaagt.integration.nnm -set OPC_SNMP_OVERRIDE_
     SEVERITY_OID<.1.3.6.1.4.1.11.2.17.19.2.2.22>
     ```

     En lo sucesivo, la gravedad del mensaje se basará en el nuevo valor de OID <*.1.3.6.1.4.1.11.2.17.19.2.2.22*>.

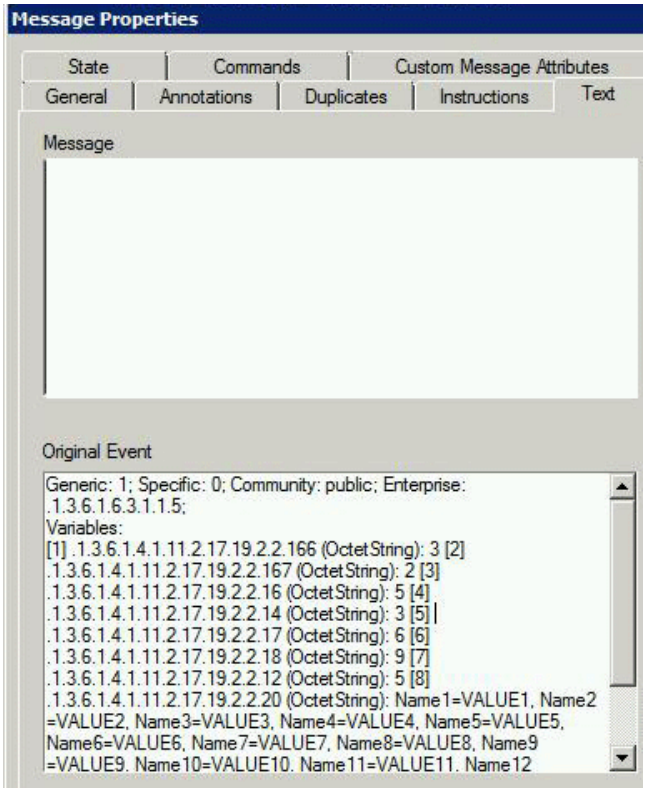
Integración con la interfaz Northbound para agregar CMA a la consola de HPOM.

Cuando una captura de SNMP se reenvía en una integración Northbound de NNMi, la captura incluye un enlace de variable que representa los atributos de

incidentes personalizados (CIA). Los CIAs de NNMi aparecen en un único atributo de mensajes de cliente (CMA) en la consola HPOM si hay un CMA[1] están configurados en la directiva HPEOM. La lista aparece con el nombre de CMA y el par de valor. Este es el comportamiento predeterminado. Los mensajes de la ficha CMA aparecen como:



Si los CMA no están configurados en la directiva HPOM, el mensaje de CIA de NNMi aparece como un mensaje de texto:



[1]. Un atributo de mensaje personalizado (CMA) puede ser cualquier información que sea significativa y puede tener más de un CMA anexado a un único mensaje.

Después de configurar **opctrapi**, el interceptor de capturas de SNMP debe acceder a los CIA y representar cada CIA como un atributo de mensaje personalizado (CMA) en el mensaje OM.

Configuración del interceptor de capturas de SNMP para crear CMA desde CIA de NNMi en la consola de HPOM

Puede configurar opctrapi y mostrar el CIA de NNMi como nombre de CMA y par de valor en la ficha de propiedades de CMA configurando lo siguiente:

- OPC_SPLIT_NNM_CUSTOM_ATTR: De manera predeterminada, esta propiedad está establecida en **FALSE**. Al establecer esta propiedad en **TRUE**, todos los valores de CIA de NNMi presentes en los enlaces de variable (**.1.3.6.1.4.1.11.2.17.19.2.2.20**) aparecerán como CMAs individuales.
- OPC_SPLIT_NNM_CUSTOM_ATTR_MAX: Este parámetro es **opcional**. Esta propiedad solo se habilita si la variable OPC_SPLIT_NNM_CUSTOM_ATTR se establece en TRUE. La variable define el número de atributos personalizados de NNMi que el mensaje de HPOM puede leer e interpretar. De manera predeterminada, el valor está establecido en 20. Esto significa que solo aparecerán 20 CMA por separado con sus valores respectivos y que el CIA de NNMi aparecerá en un único CMA. Se puede especificar el valor si se requiere.

Siga estos pasos para modificar el comportamiento predeterminado de **opctrapi** para crear CMAs separados de los CIA de NNMi:

1. Inicie sesión en el nodo con los privilegios necesarios.
2. En el símbolo del sistema,
   - Para habilitar **opctrapi** para leer y crear un CMA separado en la lista que aparece en el enlace de variables (**.1.3.6.1.4.1.11.2.17.19.2.2.20**), ejecute el comando siguiente:

     ```
     ovconfchg -ns eaagt.integration.nnm -set OPC_SPLIT_NNM_
     CUSTOM_ATTR TRUE
     ```

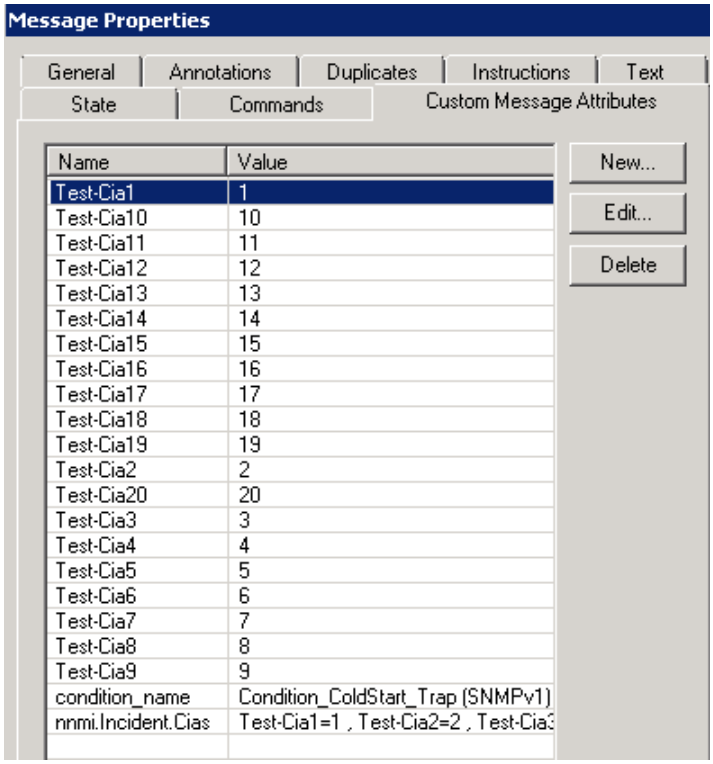
   - Para establecer el valor de la variable para leer el mensaje y crear un nombre de CMA y un par de valor separados, ejecute el comando siguiente:

     ```
     ovconfchg -ns eaagt.integration.nnm -set OPC_SPLIT_NNM_
     CUSTOM_ATTR_MAX <valor>
     ```

     En el anterior comando, <*Valor*> es un entero. De manera predeterminada, el valor es 20.

     Después de la configuración, los mensajes de la ficha CMA aparecen con los atributos del nombre y del valor. Además, el mensaje no truncado del

CIA de NNMi (`nnmi.Incident.Cias`) también aparece en la lista.



Integración con la interfaz de reenvío de capturas de NNMi para asignar el origen de la captura de SNMP

En las versiones anteriores, siempre que NNMi reenvía las capturas v2c de SNMP al nodo (Operations Agent está disponible en el nodo), el nombre del nodo de origen del mensaje HPEOM aparece como nombre del nodo NNMi.

HPE Operations Agent está integrado con la interfaz de reenvío de NNMi. La interfaz de reenvío de NNMi agrega los enlaces de variables para identificar el origen donde se originó la captura.

Puede configurar **opctrapi** para derivar el nodo de origen que generó la captura. Cuando se genera un mensaje de HPOM, el interceptor de capturas de SNMP (**opctrapi**) debe establecer el nodo de origen como el nodo donde se generó la captura y no el nombre de nodo de NNMi.

Configuración del interceptor de captura de SNMP para derivar el nombre de nodo de origen

Puede configurar la propiedad siguiente:

OPC_NODENAME_FROM_NNM_FRWD_TRAP:

De manera predeterminada, esta propiedad está establecida en **FALSE**. Al establecer la propiedad en **TRUE**, todas las capturas se buscan por los enlaces de variables ".1.3.6.1.4.1.11.2.17.2.19.1.1.2.0" and .1.3.6.1.4.1.11.2.17.2.19.1.1.3.0. **Opctrapi** usa el enlace de variable siguiente:

- (.1.3.6.1.4.1.11.2.17.2.19.1.1.2.0): para identificar el tipo de dirección IP.
- (.1.3.6.1.4.1.11.2.17.2.19.1.1.3.0): para derivar la dirección IP para establecer el nodo como el nodo de origen.

Para configurar opctrapi para que asigne el nombre de nodo de origen cuando NNMi reenvía eventos de SNMPv2, siga estos pasos:

1. Inicie sesión en el nodo con los privilegios necesarios.
2. En el símbolo del sistema, ejecute el comando siguiente para habilitar opctrapi para que lea los enlaces de variables y asigne el nodo desde donde se originó la captura:

   ```
   ovconfchg -ns eaagt.integration.nnm -set OPC_NODENAME_FROM_
   NNM_FRWD_TRAP TRUE
   ```

   Nota: Si la variable no está establecida o está establecida en False, **opctrapi** no derivará el origen (nombre de nodo) desde donde se generó la captura.

Configuración de la supresión de tormentas de mensajes

Las tormentas de mensajes ocurren cuando un número inusualmente elevado de nuevos mensajes llegan al servidor de gestión en un breve intervalo de tiempo y fluyen al explorador de mensajes activos. Este fenómeno puede conducir a interrupciones del servidor de gestión.

HPE Operations Agent puede detectar y suprimir la tormenta de mensajes en un nodo gestionado. Configure las siguientes variables:

- OPC_MSG_STORM_DETECTION - Esta variable es **obligatoria**. De manera predeterminada, esta propiedad está establecida en **FALSE**. Al establecer esta propiedad en **TRUE**, puede habilitar la detección de tormentas de mensajes.
- OPC_MSG_STORM_DETECTION_CATEGORY - Esta variable es **obligatoria**. Esta propiedad solo se habilita si la variable OPC_MSG_STORM_DETECTION se establece en TRUE. Puede establecer la variable para cualquiera de los atributos de mensaje como POLICY, MSGGROUP, APPLICATION, OBJECT o SEVERITY.

- OPC_MSG_STORM_RATE - Esta variable es **obligatoria**. La variable define los parámetros siguientes:
  - Umbral: define el límite de los mensajes entrantes. Cuando el número de mensajes entrantes excede del límite definido, se detecta una condición de tormenta de mensajes.
  - Tiempo: intervalo durante el cual se cuentan los mensajes entrantes para detectar la condición de tormenta de mensajes.
  - Restablecer: define un límite cuyo número de mensajes está por debajo del valor definido. Este parámetro se usa para detectar cuándo se resuelve la condición de tormenta.
- OPC_SEND_INTERNAL_MSG_ON_MSGSTORM - Esta variable es **opcional**. Define si se envían o se detienen los mensajes internos. De manera predeterminada, el valor se establece en TRUE.
- OPC_SUPPRESS_MSG_ON_MSG_STORM - Esta variable es **opcional**. Define si se envían o se suprimen los mensajes. El comportamiento predeterminado es que si se cumple la condición del umbral y se detecta un estado de tormenta de mensajes, se suprimirán todos los mensajes que superen el valor de umbral. De manera predeterminada, el valor es TRUE.
- OPC_MSG_STORM_TRACE_SUPPRESSED_MSGS - Esta variable es **opcional**. Define si se registran los mensajes en el archivo de registro solo cuando <OPC_SUPPRESS_MSG_ON_MSG_STORM > está establecido en TRUE. De manera predeterminada, el valor es FALSE.

Cuando el agente de mensajes detecta una tormenta de mensajes o cuando esta se resuelve, se registra un mensaje en el archivo de registro *(System.txt)*:

En Windows:

`%OvDataDir%\log`

En HP-UX/Linux/Solaris:

`/var/opt/OV/log`

Estos parámetros están disponibles en el espacio de nombres `eaagt.msgstorm`.

Las ventajas de la detección de la tormenta de mensajes en el nodo gestionado son las siguientes:

- No se requieren circuitos ECS.
- La tormenta de mensajes se identifica en el nodo de origen.
- Pasos de configuración sencillos que se pueden configurar para varios atributos de mensaje.

Nota: Ejemplo para detectar y suprimir la tormenta de mensajes en un nodo gestionado.

Puede detectar la condición de tormenta estableciendo el parámetro OPC_MSG_STORM_DETECTION en TRUE.

Después de habilitar la condición de tormenta de mensajes, defina los parámetros siguientes:

OPC_MSG_STORM_DETECTION_CATEGORY como POLICY. Las directivas desplegadas son Opclepolicy y Opcmsgipolicy.

Para establecer el parámetro OPC_MSG_STORM_RATE, puede calcular la frecuencia de mensajes entrantes para establecer los parámetros. Consulte "Configuración de la supresión de tormentas de mensajes" en la página 51. Puede establecer los valores en función de la frecuencia de mensajes.

Establezca los valores del parámetro OPC_MSG_STORM_RATE como Umbral = 100, Tiempo = 20 y Valor de restablecimiento = 50

Opclepolicy envía 50 mensajes y Opcmsgipolicy envía 101 mensajes. La tormenta que se detecta cuando hay 101 mensajes desde Opclepolicy, frente al valor del umbral establecido (100).

Puede obtener la notificación de cuando se detecta la tormenta de mensajes y se resuelve mediante el parámetro OPC_SEND_INTERNAL_MSG_ON_MSGSTORM. De manera predeterminada, el valor se establece en TRUE.

De manera predeterminada, el recuento de mensajes por encima de 100 se suprimirán o ignorarán. Puede establecer el parámetro (OPC_SUPPRESS_MSG_ON_MSG_STORM) to FALSE para obtener los mensajes que se están suprimiendo.

Puede guardar los mensajes suprimidos en el archivo de registro establecimiento el parámetro (OPC_MSG_STORM_TRACE_SUPPRESSED_MSGS).

Configuración de la detección y supresión de tormentas de mensajes

Nota: Asegúrese de reiniciar el agente de mensajes siempre que cambie cualquier parámetro para la configuración. Si el agente de mensajes se reinicia, el contenido asociado con la detección de tormentas de mensajes se restablece.

Para configurar la detección o supresión de tormentas de mensajes, siga estos pasos:

1. Inicie sesión en el nodo con los privilegios necesarios o configúrelos remotamente.
2. Para habilitar la detección de tormenta de mensajes, ejecute el comando siguiente:

   ```
   ovconfchg -ns eaagt.msgstorm -set OPC_MSG_STORM_DETECTION TRUE
   ```

   El valor predeterminado se establece en FALSE.
3. Para establecer la categoría del mensaje, ejecute el comando siguiente:

   ```
   ovconfchg -ns eaagt.msgstorm -set OPC_MSG_STORM_DETECTION_
   CATEGORY<categoría>
   ```

   Nota: Es una variable obligatoria.

   La categoría debe definirse como **ona** de las siguientes:
   - POLICY
   - GROUP
   - APPLICATION
   - OBJECT
   - SEVERITY

   Nota: Solo puede definir una de las opciones de variables. La combinación de los distintos valores no está disponible.
4. Ejecute el comando para establecer las variables de la frecuencia de mensajes.

   ```
   ovconfchg -ns eaagt.msgstorm -set OPC_MSG_STORM_
   RATE<ThresholdValue:Time:ResetValue>
   ```

   Nota: Estas variables son obligatorias.

   Las variables se definen del siguiente modo:
   - Valor de umbral: establezca un valor numérico. Si el número de mensajes en el intervalo de tiempo establecido es mayor que el valor de umbral, la condición se conoce como condición de tormenta de mensajes. En esta condición, el agente de mensajes suprime los mensajes hasta que se cumpla la condición de restablecimiento. El valor de umbral se marca para

el estado de tormenta de mensajes para todos los grupos disponibles en la categoría establecida. Puede comprobar la frecuencia de mensajes para establecer este parámetro. Consulte "Configuración de la detección y supresión de tormentas de mensajes " en la página 53.

Ejemplo: La categoría se establece en SEVERITY. En esta categoría de gravedad de mensajes, hay cinco tipos de grupos como crítico, grave, leve, aviso y normal. Los mensajes se agrupan en función de la gravedad y se comprueba el valor de umbral configurado.

Establezca el valor de umbral en 100. Los mensajes recibidos para la gravedad crítica, grave, leve, aviso y normal son 80, 60, 40, 50 y 110, respectivamente. Aquí la tormenta de mensajes se detecta solo para el estado normal, ya que el número de mensajes (110) es mayor que el valor de umbral establecido (100). Los 10 mensajes restantes de gravedad normal se suprimirán.

- Tiempo: establezca el intervalo de tiempo en segundos. El intervalo de tiempo en el que se registra el número de mensajes entrantes. El valor recomendado es menor de 900 segundos.
- Valor de restablecimiento: establezca un valor numérico que sea menor o igual que el valor de umbral. La detección de tormenta de mensajes se restablece si se cumplen las condiciones siguientes:
  - La frecuencia de mensajes se comprueba para el intervalo de tiempo especificado.
  - La frecuencia de mensajes debe ser menor que el valor de restablecimiento.

Ejemplo para establecer los parámetros para la condición de tormenta de mensajes.

set OPC_MSG_STORM_DETECTION=TRUE

set OPC_MSG_STORM_DETECTION_CATEGORY=SEVERITY

set OPC_MSG_STORM_RATE=100:60:45

donde el valor de umbral es 100, el tiempo es 60 y el valor de restablecimiento es 45.

Después de la detección de tormenta de mensajes, el número de mensajes entrantes se comprueban en el intervalo periódico establecido (60 segundos). En el intervalo de tiempo establecido, si los mensajes son

inferiores al valor de restablecimiento (45), el agente de mensajes (opcmsga) detiene la omisión de los mensajes y finaliza el estado de tormenta de mensajes.

Nota: Los pasos 5, 6 y 7 son pasos opcionales.

5. Para recibir y detener los mensajes internos, ejecute el comando siguiente:

   `ovconfchg -ns eaagt.msgstorm -set OPC_SEND_INTERNAL_MSG_ON_MSGSTORM<valor>`

   El <*Valor*> debe definirse como *uno* de los siguientes:

   - TRUE: los mensajes internos se generan siempre que el estado de tormenta de mensajes se detecte y el estado se resuelva. El valor predeterminado es TRUE.
   - FALSE: no se generan mensajes internos para el estado de tormenta de mensajes y siempre que el estado se resuelva.

6. En la condición de tormenta de mensajes, ejecute el comando siguiente para suprimir o recibir los mensajes:

   `ovconfchg -ns eaagt.msgstorm -set OPC_SUPPRESS_MSG_ON_MSG_STORM<valor>`

   El valor debe definirse como *uno* de los siguientes:

   - TRUE: el agente de mensajes suprimirá los mensajes en la condición de tormenta de mensajes. El valor predeterminado es TRUE.
   - FALSE: el agente de mensajes no suprimirá los mensajes en la condición de tormenta de mensajes.

7. **Omita este paso** si la variable OPC_SUPPRESS_MSG_ON_MSG_STORM se establece en FALSE.

   En la condición de tormenta de mensajes, ejecute el comando siguiente para registrar los mensajes suprimidos:

   `ovconfchg -ns eaagt.msgstorm -set OPC_MSG_STORM_TRACE_SUPPRESSED_MSGS <valor>`

   El valor debe definirse como *uno* de los siguientes:

   - TRUE: el agente de mensajes guardará todos los mensajes suprimidos en el archivo de registro.

- FALSE: el agente de mensajes no guardará los mensajes suprimidos. De manera predeterminada, el valor es FALSE.

Los mensajes suprimidos están disponibles en (*msgsuppress.log <ID de proceso del proceso>*) en el siguiente directorio:

En Windows:

```
%OvDataDir%\tmp\OpC
```

En HP-UX/Linux/Solaris:

```
/var/opt/OV/tmp/OpC
```

8. Ejecute el comando siguiente para reiniciar el agente de mensajes:

```
ovc –restart opcmsga
```

Comprobación de la frecuencia de mensajes

La frecuencia de mensajes se usa para medir la frecuencia media de los mensajes por el sistema. Después de comprobar la frecuencia media de mensajes, puede configurar un límite de umbral ideal y restablecer el límite para detectar y suprimir la condición de tormenta de mensajes. Solo puede comprobar la frecuencia de mensajes después de configurar la detección de tormenta de mensajes.

Para comprobar la frecuencia de mensajes en un período de tiempo concreto, ejecute el comando siguiente:

En Windows:

```
%OVInstallDir%/lbin/eaagt/opcmsga –message_rate
```

En HP-UX/Linux/Solaris:

```
/opt/OV/lbin/eaagt/opcmsga –message_rate
```

En AIX:

```
/usr/lpp/OV/lbin/eaagt/opcmsga –message_rate
```

Configuración del servidor de copia de seguridad

El agente de mensajes envía los mensajes al servidor principal, que es su HP Operations Manager (HPOM). Si tiene más de un servidor HPOM en el entorno, también puede configurar otro servidor para que actúe como el servidor de copia de seguridad en el entorno. Cuando tenga un servidor de copia de seguridad configurado en el entorno, si el vínculo de comunicación con el servidor principal está inactivo y los mensajes no se pueden enviar al servidor principal, el agente de mensajes envía los mensajes al servidor de copia de seguridad.

También puede configurar el servidor de copia de seguridad para la distribución de carga de la sincronización de los mensajes entre los servidores principal y el de copia de seguridad (por ejemplo, en el escenario gestor-de-gestor (MOM)). Para obtener más información sobre el escenario MOM, consulte la guía *HP Operations Manager for UNIX Concepts Guide* o la *ayuda en línea de HP Operations Manager*. Un servidor de copia de seguridad puede ser otro servidor HPOM de su entorno. Puede configurar uno o varios servidores como servidor de copia de seguridad.

Puede habilitar HPE Operations Agent para que envíe mensajes al servidor de copia de seguridad configurando los valores para las variables siguientes:

- OPC_BACKUP_MGRS: puede especificar un servidor o una lista de servidores para ser configurados como servidores de copia de seguridad. Los valores de la lista se pueden separar por comas o puntos y coma.
- OPC_BACKUP_MGRS_FAILOVER_ONLY: cuando establece el valor para esta variable en **TRUE**, el agente de mensajes reenvía los mensajes a los servidores de copia de seguridad solo si el servidor principal está inactivo. Y cuando el valor se establece en **FALSE**, los mensajes se envían a los servidores de copia de seguridad, con independencia del estado del servidor principal. El valor predeterminado de esta variable es **FALSE**.

Si ha configurado la lista de servidores de copia de seguridad, durante la inicialización, el agente de mensajes crea una lista de los servidores de copia de seguridad y después busca el valor establecido para la variable OPC_BACKUP_MGRS_FAILOVER_ONLY. Cuando llega un mensaje al archivo `msgagtdf` y el valor de la variable OPC_BACKUP_MGRS_FAILOVER_ONLY es:

- **FALSE**- El agente de mensajes envía el mensaje al servidor principal y después al servidor de copia de seguridad. Cuando el mensaje se envía correctamente a ambos servidores, las entradas del mensaje se suprimen del archivo `msgagtdf`.
- **TRUE**- El agente de mensajes envía los mensajes al servidor principal. Si se produce un error al enviar el mensaje, el agente de mensajes intenta reenviar el mensaje a los servidores de copia de seguridad. Si hay más de un servidor de copia de seguridad enumerado y el mensaje se entrega al menos en uno de ellos, la entrada del mensaje se suprime del archivo `msgagtdf`. El agente de mensajes no intenta reenviar el mensaje a los otros servidores de copia de seguridad y tampoco envía el mensaje al servidor principal, cuando está funcionando de nuevo.

Nota: Cuando OPC_BACKUP_MGRS_FAILOVER_ONLY está establecido en **TRUE**, si un mensaje que inicia una acción automática local se envía al servidor principal y después este servidor está inactivo, el agente de mensajes

envía la respuesta de la acción al gestor de copia de seguridad. Pero lo descarta el gestor de copia de seguridad ya que el servidor de copia de seguridad no tiene el mensaje original que inció la acción.

Requisitos previos:

- Deben instalarse los certificados de confianza de los servidores de copia de seguridad en el nodo de HPE Operations Agent.
- Debe instalarse el certificado de confianza del servidor principal en el servidor de copia de seguridad.
- Debe agregarse al nodo de HPE Operations Agent a los servidores de copia de seguridad.

Para establecer los valores de la variables, siga estos pasos:

1. Inicie sesión en el nodo como administrador o usuario raíz.
2. Ejecute los comandos siguientes:
   - Para establecer el servidor de copia de seguridad, ejecute el siguiente comando:

     `ovconfchg -ns eaagt -set OPC_BACKUP_MGRS` *<nombre_de_servidor>*

     <nombre_de_servidor> es el nombre del servidor de copia de seguridad. Por ejemplo, *abc.ind.hp.com*.

     Para configurar la lista de servidores de copia de seguridad, utilice cualquiera de estos comandos:

     - `ovconfchg -ns eaagt -set OPC_BACKUP_MGRS` *<nombre_de_servidor1>,<nombre_de_servidor2>,...,<nombre_de_servidorN>*
     - `ovconfchg -ns eaagt -set OPC_BACKUP_MGRS` *<nombre_de_servidor1>;<nombre_de_servidor2>;...;<nombre_de_servidorN>*

       La secuencia de los servidores de copia de seguridad dependen de la secuencia en la que especifique los nombres del servidor. En los ejemplos anteriores, *<nombre_de_servidor1>* actúa como el primer servidor de copia de seguridad y *<nombre_de_servidor2>* como el segundo servidor de copia de seguridad.

   - Para modificar el valor de la variable OPC_BACKUP_MGRS_FAILOVER_ONLY, ejecute el comando siguiente:

     `ovconfchg -ns eaagt -set OPC_BACKUP_MGRS_FAILOVER_ONLY TRUE`

     El valor predeterminado es **FALSE**.

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 HPE Operations Agent.

Comprobación de la licencia para el proceso Perfd

Este componente sólo puede utilizarse en combinación con HP Ops OS Inst to Realtime Inst LTU, Glance Pak Software LTU o Glance Software LTU. Para obtener más información, consulte la guía *HPE Operations Agent License Guide*.

El proceso **perfd** inicia la recopilación de datos solo después de comprobar que la licencia está habilitada. Si no lo está, el proceso está inactivo. Todas las actividades se registran en los archivos de registro.

Modificación de la configuración

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 HP-UX/Linux/Solaris:

```
/var/opt/perf
```

Parámetro	Descripción	Valor predeterminado
interval	La frecuencia de la recopilación de datos en segundos. Este valor debe ser un múltiplo o factor de 60.	10
puerto	El puerto usado por **perfd**.	5227
depth	El período de tiempo en el que se conservan los valores de métricas globales en la caché de **perfd**.	30

, continuación

Parámetro	Descripción	Valor predeterminado
	Estos datos se utilizan para el resumen de datos.	

maxrps	El número máximo de solicitudes de sesión por segundo aceptadas por perfd. Si el número de solicitudes excede el límite, **perfd** hace una pausa de un segundo y, a continuación, registra los detalles de este evento en el archivo de registro. El archivo de registro, `status-perfd.<puerto>`, está ubicado en el directorio siguiente del nodo:
	En Windows:
	`%ovdatadir%`
	En HP-UX/Linux/Solaris:
	`/var/opt/perf`
		20

maxtpc	El número máximo de sesiones por sistema de cliente aceptadas por **perfd**. Cuando el número de disponible de sesiones alcanza este límite, si llega una solicitud adicional, **perfd** la deniega.	30
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	False

, continuación

Parámetro	Descripción	Valor predeterminado
	para los procesos, aplicación, operaciones NFS, sistemas lógicos y ARM. Además, no se recopilarán los datos HBA y LVM en HP-UX.	
localonly	Si se establece en **true**, **perfd** solo 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 IPv6. De manera predeterminada, si **perfd** no puede crear un socket IPv6, cambia automáticamente al socket de IPv4. **Nota:** Si desea explícitamente que **perfd** acepte solo conexiones IPv4, establezca el valor de IPv4 en true en el archivo **perfd.ini**.	False
logsize	Este parámetro especifica un tamaño de archivo sobre el cual **perfd** sustituirá sus archivos de registro o de rastreo. Si especifica un tamaño inferior a 4096 bytes, se ignora.	1048576 bytes

, continuación

Parámetro	Descripción	Valor predeterminado
add	Use este parámetro para especificar una lista de clases de métricas separadas por comas para las que los datos deben recopilarse. Este parámetro es útil para agregar clases de métricas individuales cuando el parámetro ligero se establece en True.	
exclude	Use este parámetro para especificar una lista de clases de métricas separadas por comas para las que los datos no deben recopilarse. Este parámetro se usa para excluir clases de métricas individuales cuando el parámetro ligero es 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 HPE Operations Agent para que los cambios surtan efecto.

Monitorización de HPE Operations Agent en las conexiones IPv6

Cuando haya finalizado la instalación de HPE Operations Agent 12.01, determina la configuración de IP admitida y enlaza la dirección IP correspondiente. No se requiere la configuración adicional para las conexiones IPv6. Para un nodo de doble pila, el nodo determina la dirección IP del servidor admitido.

Servidor	Nodo de doble pila
IPv4	usa IPv4
IPv6	usa IPv6
Doble pila	usa IPv6

Nota: La comunicación de IPv6 se prefiere si tanto el servidor como el nodo son de doble pila.

Restricción de acceso

HPE Operations Agent se puede configurar para que impida el acceso de otros sistemas a los datos de rendimiento en tiempo real del sistema local utilizando el componente RTMA.

Las utilidades **cpsh**, **padv** y **mpadv** permiten acceder de forma remota a los datos de rendimiento en tiempo real del nodo del agente. La utilidad **cpsh** se puede usar para ver los datos de rendimiento en tiempo real en todos los sistemas remotos en los que se ejecuta el proceso **perfd**.

Las utilidades **padv** o **mpadv** se pueden usar para ejecutar scripts de **asesor** en todos los sistemas remotos en los que se ejecuta el proceso **perfd**.

Estas utilidades dependen del proceso **perfd** para acceder a los datos de rendimiento en tiempo real.

Para impedir que otros sistemas accedan a los datos de rendimiento en tiempo real del sistema local revocando su acceso al proceso de perfd, siga estos pasos:

Nota: Este procedimiento no impide que la vista de diagnóstico de HP Performance Manager acceda a los datos de rendimiento en tiempo real desde el sistema.

1. Inicie una sesión en el sistema como administrador o usuario raíz. Vaya al directorio siguiente:

   En Windows:

   `%OvDataDir%\`

   En HP-UX/Linux/Solaris:

   `/var/opt/perf/`

2. Ahora, los restantes sistemas del entorno (donde HPE Operations Agent está disponible) no pueden acceder a los datos de rendimiento en tiempo real desde este sistema.

3. Cree un archivo vacío en esta ubicación y guárdelo **authip**.

4. Para permitir que un sistema acceda a los datos de rendimiento en tiempo real desde este sistema, abra el archivo **`authip`** con un editor de texto, agregue el nombre de dominio completo o la dirección IP del sistema y guarde el archivo.

Puede especificar varios nombres de dominio completo o direcciones IP (las entradas se deben separar con líneas nuevas).

5. Reinicie **perfd** para habilitar los cambios.

Los sistemas que se especifican en el archivo **authip** ahora pueden acceder a los datos de rendimiento en tiempo real desde este sistema.

6. Para permitir el acceso a todos los sistemas, elimine el archivo **authip**.

Configuración del usuario del agente

HPE 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, es posible configurar HPE Operations Agent para que se ejecute con un usuario que no sea el predeterminado y que tenga menos privilegios que el usuario raíz o el usuario Sistema local.

Puede ejecutar solo Componente Operations Monitoring con un usuario que no sea el raíz ni el sistema local y los restantes componentes con el usuario raíz /Sistema local predeterminado.

En función de la cuenta de usuario que se utilice, puede configurar los siguientes modos de funcionamiento del agente:

- **Sin privilegios:** Todos los componentes de HPE Operations Agent se ejecutan con una cuenta de usuario que no el predeterminado y que tiene menos privilegios que el usuario raíz o Sistema local.

  Nota: En HP-UX, HPE Operations Agent no se puede ejecutar en el modo sin privilegios.

  No se debe ejecutar HPE Operations Agent en un modo de usuario sin privilegios cuando HPE Operations Agent y NNMi coexisten en un sistema.

- **Raíz:** Todos los componentes de HPE Operations Agent se ejecutan con el usuario predeterminado (raíz o Sistema local). Es el modo de funcionamiento predeterminado de Agent.
- **Mixto:** Componente Performance Collection se ejecuta con el usuario raíz y Componente Operations Monitoring se ejecuta con la cuenta de usuario local.

  Nota: Al ejecutarse en modo mixto, se recomienda que el usuario raíz o con privilegios inicie los procesos de Agent.

Cuando Agent está configurado para ejecutarse bajo un usuario no predeterminado en un sistema en el que está instalado HP Performance Manager,

el servicio `OvTomCatB` de HP Performance Manager empieza a ejecutarse bajo un usuario de Agent que no sea el predeterminado.

En los entornos gestionados por HPOM, adicionalmente se puede configurar Agent para que ejecute comandos automáticos o iniciados por un operador con un usuario que no sea el usuario bajo el que se ejecuta.

Requisitos para utilizar un usuario no predeterminado

El usuario de Agent que se desea usar debe cumplir los requisitos siguientes:

- **Requisitos exclusivos de Windows:**
  - El usuario debe tener control total de la clave del Registro `HKEY_LOCAL_MACHINE/Software/Hewlett-Packard/OpenView`
  - El usuario debe tener acceso de lectura a la clave del Registro `HKEY_LOCAL_MACHINE/Software/Microsoft/WindowsNT/CurrentVersion/Perflib`
  - El usuario debe tener derechos para:
    - Iniciar sesión como un servicio - Para iniciar y detener los servicios de Componente Performance Collection.
    - Gestionar registros de auditoría y seguridad - El usuario puede ver los eventos auditados en el registro de seguridad del visor de eventos. Un usuario con este privilegio puede ver y borrar el registro de seguridad.
    - Actuar como parte del sistema operativo - Para controlar los procesos del agente.
    - Reemplazar un token de nivel de proceso - Para crear los procesos del agente como el usuario no raíz.
- Para monitorizar un archivo de registro con una directiva, el usuario del agente debe tener permiso para leer dicho archivo de registro.

  Nota: En Windows, el usuario del agente no puede leer el archivo de registro ya que el usuario no tiene permiso.
- Para 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.

Limitaciones en el uso de un usuario no predeterminado

El usuario del agente no predeterminado (no raíz y sin privilegios) que se desea usar tiene las limitaciones siguientes:

- El modo de usuario sin privilegios no se admite en la plataforma HP-UX.
- El modo de usuario sin privilegios no se admite en el servidor de gestión HPOM, pero el modo de usuario mixto se admite en HP Operations Manager para UNIX, Linux y Solaris desde la versión 9.20.
- Los modos Mixto y Usuario sin privilegios no se admiten en AIX WPAR.
- Los datos de métricas BYLS no pueden recopilarse para Xen, KVM, VMware vSphere, Hyper-V y otros dominios de virtualización.
- De manera predeterminada, el usuario de agente con modos de usuario sin privilegios y mixtos no tendrán permiso para leer el archivo de registro monitorizado.
- De manera predeterminada, el usuario de agente con modos de usuario mixtos y sin privilegios no tendrán permiso para iniciar un programa usando un comando automático, un comando iniciado por operador, una herramienta programada.
- HP Operations Smart Plug-ins puede requerir una configuración adicional o derechos de usuario si el usuario de agente con modos sin privilegio y mixto no dispone de derechos administrativos.
- El HPE Operations Agent no puede recopilar las métricas que comienzan por `PROC_REGION_*` o `PROC_FILE_*` para todas las instancias de procesos poseídos por usuarios sin privilegios. Además, dependiendo del sistema operativo, los procesos que se ejecutan con privilegios más elevados, como **ovbbccb** y `sshd` estarán disponibles o no disponibles en el modo sin privilegios.
- En Windows, la métrica `PROC_USER_NAME` se muestra como `Desconocida` en procesos que posean usuarios que no sean el usuario de Agent.
- En AIX, puede ver el siguiente mensaje de error en la consola de la línea de comandos (o en el archivo `oainstall.log` del directorio`/var/opt/OV/log`) después de configurar Agent para que use un usuario que no sea el predeterminado:

  ```
  Product activation failure. Refer to the log file for
  moredetails.
  ```

  Ignore este error.
- En Solaris, HPE Operations Agent obtiene los detalles del proceso solo hasta 80 caracteres, lo que es una limitación de Solaris. **opcmona** lee el archivo

`/proc/pid/psinfo` y almacena los resultados en una estructura. Si la información extendida se requiere más adelante, el sistema lee el archivo `/proc/pid/as`. Si el agente se está ejecutando como un usuario no raíz y no tiene permiso para abrir el archivo `/proc/pid/as`, compara los detalles del proceso con la información limitada disponible en `psinfo`.

Nota: Para configurar el agente durante la instalación para que se ejecute en un usuario no predeterminado, consulte la *Guía de instalación de HPE Operations Agent y HPE Operations Smart Plug-ins para Infrastructure*.

Configurar el usuario de Agent después de la instalación

Si no puede configurar HPE Operations Agent para que utilice un usuario no predeterminado en el momento de la instalación, puede usar la opción `-configure` de la secuencia de comandos `oainstall` (que está disponible en el nodo del agente) o el comando `ovswitchuser` después de la instalación para finalizar esta configuración.

Cambio del usuario predeterminado en Windows

Si no puede configurar el agente para que se ejecute con un usuario no predeterminado en el momento de la instalación (consulte Configure the Agent User During Installation), se recomienda instalar HPE Operations Agent en el modo *inactivo*. Para obtener más información, consulte la guía *HPE Operations Agent and HPE Operations Smart Plug-ins for Infrastructure Installation Guide*.

Puede usar cualquiera de los siguientes métodos para configurar Agent para que utilice un usuario que no sea el predeterminado:

- Usar un archivo de perfil
- Use el comando ovswitchuser

Método alternativo: Use el comando ovswitchuser

Nota: Asegúrese de detener los procesos de Operations Agent antes de comenzar a usar el comando `ovswitchuser.vbs`.

Si no desea usar un archivo de perfil, siga estos pasos:

Nota: La configuración con el archivo de perfil es el procedimiento de configuración recomendado.

1. Detenga Agent:

   En los nodos de Windows 64 bits:

   ```
   %ovinstalldir%bin\win64\opcagt -kill
   ```

   En otros nodos de Windows:

   ```
   %ovinstalldir%bin\opcagt -kill
   ```

2. Ejecute el siguiente comando para configurar Agent para que se ejecute con un usuario no predeterminado:

   ```
   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.

   Nota: 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.

3. Ejecute el siguiente comando para establecer los permisos necesarios para el usuario no predeterminado:

   ```
   cscript %ovinstalldir%lbin\xpl\ovsetscmpermissions.vbs -
   user<Nombre de usuario> -f
   ```

   En este caso, <*Nombre de usuario*> es el nombre del usuario no predeterminado.

4. Ejecute uno de los siguientes comandos:

   Para ejecutar todos los componentes de Agent en una cuenta que no sea sistema local:

   En los nodos de Windows 64 bits:

   ```
   %OvInstallDir%bin\win64\ovconfchg -ns eaagt -set MODE NPU
   ```

   En los restantes nodos de Windows:

   ```
   %OvInstallDir%bin\ovconfchg -ns eaagt -set MODE NPU
   ```

Para ejecutar sólo Componente Operations Monitoring con una cuenta que no sea sistema local:

En los nodos de Windows 64 bits:

```
%OvInstallDir%bin\win64\ovconfchg -ns eaagt -set MODE MIXED
```

En los restantes nodos de Windows:

```
%OvInstallDir%bin\ovconfchg -ns eaagt -set MODE MIXED
```

5. Ejecute los comandos siguientes:

   En los nodos de Windows 64 bits:

   ```
   %OvInstallDir%bin\win64\ovconfchg -ns ctrl.sudo -set OV_SUDO_
   USER<Nombre de usuario>
   ```

   ```
   %OvInstallDir%bin\win64\ovconfchg -ns ctrl.sudo -set OV_SUDO_
   GROUP<Nombre de grupo>
   ```

   ```
   %OvInstallDir%bin\win64\ovconfchg -ns eaagt -set OPC_PROC_
   ALWAYS_INTERACTIVE NEVER
   ```

   En los restantes nodos de Windows:

   ```
   %OvInstallDir%bin\ovconfchg -ns ctrl.sudo -set OV_SUDO_
   USER<Nombre de usuario>
   ```

   ```
   %OvInstallDir%bin\ovconfchg -ns ctrl.sudo -set OV_SUDO_
   GROUP<Nombre de grupo>
   ```

   ```
   %OvInstallDir%bin\ovconfchg -ns eaagt -set OPC_PROC_ALWAYS_
   INTERACTIVE NEVER
   ```

   En esta instancia, *<Nombre de usuario>* es el nombre del usuario que no es el predeterminado y *<Nombre de grupo>* es el grupo al que pertenece el usuario no predeterminado.

6. Si ha elegido el modo de funcionamiento Sin privilegios, ejecute el siguiente comando:

   En los nodos de Windows 64 bits:

   ```
   %OvInstallDir%bin\win64\ovconfchg -ns eaagt -set OPC_RPC_ONLY
   TRUE
   ```

   En los restantes nodos de Windows:

   ```
   %OvInstallDir%bin\ovconfchg -ns eaagt -set OPC_RPC_ONLY TRUE
   ```

7. Si ha elegido el modo de funcionamiento Mixto, ejecute el siguiente comando:

   En los nodos de Windows 64 bits:

   ```
   %OvInstallDir%bin\win64\ovconfchg -ns eaagt -set NPU_TASK_SET
   EVENT_ACTION
   ```

   En los restantes nodos de Windows:

```
%OvInstallDir%bin\ovconfchg -ns eaagt -set NPU_TASK_SET EVENT_
ACTION
```

8. Ejecute el comando siguiente:

   Nota: Éste es un requisito para los nodos de UNIX/Linux, pero no es un paso obligatorio para los nodos de Windows. No obstante, se recomienda realizarlo también en estos últimos.

   En los nodos de Windows 64 bits:

   `%OvInstallDir%bin\win64\ovconfchg -ns eaagt -set SNMP_TRAP_PORT` *<Puerto SNMP>*

   `%OvInstallDir%bin\win64\ovconfchg -ns bbc.cb -set SERVER_PORT`*<Puerto de comunicaciones>*

   En los restantes nodos de Windows:

   `%OvInstallDir%bin\ovconfchg -ns eaagt -set SNMP_TRAP_PORT` *<Puerto SNMP>*

   `%OvInstallDir%bin\ovconfchg -ns bbc.cb -set SERVER_PORT`*<Puerto de comunicaciones>*

   En este ejemplo:

   <Puerto de comunicaciones> es el número de puerto de comunicaciones que prefiera. Si establece SERVER_PORT en 383, no se requiere configuración adicional. Si no establece el valor del puerto en 383, asegúrese de configurar el puerto en el servidor de gestión HPOM.

   <Puerto SNMP> es el puerto en el que HPE Operations Agent recibe capturas de SNMP.

   Estos puertos deben estar por encima del 1024.

9. Reinicie el agente:

   En los nodos de Windows 64 bits:

   ```
   %ovinstalldir%bin\win64\opcagt -start
   ```

   En los restantes nodos de Windows:

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

Cambio del usuario predeterminado en UNIX/Linux

Si no puede configurar el agente para que se ejecute con un usuario no predeterminado en el momento de la instalación (consulte Configure the Agent User During Installation), se recomienda instalar HPE Operations Agent en el modo *inactivo*. Para obtener más información, consulte la guía *HPE Operations Agent and HPE Operations Smart Plug-ins for Infrastructure Installation Guide*.

Puede usar cualquiera de los siguientes métodos para configurar Agent para que utilice un usuario que no sea el predeterminado:

- Usar un archivo de perfil
- Use el comando ovswitchuser

Usar un archivo de perfil

Para cambiar el usuario predeterminado de Agent con ayuda de un archivo de perfil, siga estos pasos:

1. En el sistema en que desee instalar Agent, cree un archivo y ábralo con un editor de texto.
2. Escriba una de las siguientes instrucciones para especificar el modo en que va a funcionar Agent:

   Para ejecutar Agent en una cuenta que no sea sistema local, escriba:

   set eaagt:MODE=NPU

   Para ejecutar solo Componente Operations Monitoring con una cuenta que no sea sistema local, escriba:

   set eaagt:MODE=MIXED
3. Si ha seleccionado el modo Sin privilegios (es decir, si ha escrito `set eaagt:MODE=NPU` en el paso anterior), escriba la siguiente instrucción:

   set eaagt:OPC_RPC_ONLY=TRUE
4. Si ha seleccionado el modo Sin privilegios (es decir, si ha escrito `set eaagt:MODE=NPU` en el paso anterior), escriba las siguientes instrucciones:

   set eaagt:SNMP_TRAP_PORT=<*Número_puerto_SNMP*>

   set bbc.cb:SERVER_PORT=<*Número_puerto_comunicaciones*>

   Nota: Éste es un requisitos para los nodos de UNIX/Linux, ya que en UNIX/Linux el usuario no raíz no tiene permiso para acceder a puertos por debajo del 1024.

   En este ejemplo:

   <*Puerto de comunicaciones*> es el número de puerto de comunicaciones que prefiera.

   <*Puerto SNMP*> es el puerto en el que HPE Operations Agent recibe capturas de SNMP.

   Estos puertos deben estar por encima del 1024.
5. Si ha seleccionado el modo mixto (es decir, si ha escrito `set eaagt:MODE=MIXED` en el paso 2), escriba la siguiente instrucción:

set eaagt:NPU_TASK_SET=EVENT_ACTION

6. Escriba las siguientes instrucciones:

   set ctrl.sudo:OV_SUDO_USER=<*Nombre_de_usuario*>

   set ctrl.sudo:OV_SUDO_GROUP=<*Nombre_grupo*>

   En esta instancia, <*Nombre de usuario*> es el nombre del usuario que no es el predeterminado y <*Nombre de grupo*> es el grupo al que pertenece el usuario no predeterminado.
7. Guarde el archivo en un directorio local del sistema.
8. Vuelva a configurar Agent para que se ejecute con el usuario especificado en el archivo de perfil:
   a. Vaya a la siguiente ubicación en el nodo:

      `/opt/OV/bin/OpC/install`

   b. Ejecute el comando siguiente:

      `./oainstall.sh -a -configure -agent_profile`<*ruta_acceso*>/<*archivo_perfil*>

      En este caso, <*archivo_perfil*> es el nombre del archivo de perfil y <*ruta_acceso*> es la ruta de acceso completa al archivo de perfil.

Método alternativo: Use el comando ovswitchuser

Nota: Asegúrese de detener los procesos de Operations Agent antes de comenzar a usar el comando **ovswitchuser.sh**.

Si no desea usar un archivo de perfil, siga estos pasos:

Nota: La configuración con el archivo de perfil es el procedimiento de configuración recomendado.

1. Vaya al directorio siguiente:

   En HP-UX/Linux/Solaris:

   `/opt/OV/bin`

   En AIX:

   `/usr/lpp/OV/bin`
2. Ejecute el comando siguiente para detener el agente:

   `./opcagt -kill`
3. Ejecute el siguiente comando para configurar Agent para que se ejecute con un usuario no predeterminado:

`./ovswitchuser.sh -existinguser`*<Nombre de usuario>*`-existinggroup`*<Nombre de grupo>*

En este ejemplo:

<Nombre de usuario> es el nombre del usuario en el que se ejecuta Agent.

<Nombre de 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.

Nota: 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. Ejecute uno de los siguientes comandos:

   Para ejecutar todos los componentes de Agent en un usuario que no sea raíz:

   `%OvInstallDir%bin\ovconfchg -ns eaagt -set MODE NPU`

   Para ejecutar sólo Componente Operations Monitoring con un usuario no raíz:

   `%OvInstallDir%bin\ovconfchg -ns eaagt -set MODE NPU`

5. Ejecute el comando siguiente:

   Nota: Éste es un requisitos para los nodos de UNIX/Linux, ya que en UNIX/Linux el usuario no raíz no tiene permiso para acceder a puertos por debajo del 1024.

   `./ovconfchg -ns eaagt -set SNMP_TRAP_PORT`*<Puerto SNMP>*

   `./ovconfchg -ns bbc.cb -set SERVER_PORT`*<Puerto de comunicaciones>*

   En este ejemplo:

   <Puerto de comunicaciones> es el número de puerto de comunicaciones que prefiera. Si establece SERVER_PORT en 383, no se requiere configuración adicional. Si no establece el valor del puerto en 383, asegúrese de configurar el puerto en el servidor de gestión HPOM.

<Puerto SNMP> es el puerto en el que HPE Operations Agent recibe capturas de SNMP.

Estos puertos deben estar por encima del 1024.

6. Ejecute los siguientes comandos para iniciar Agent:

```
./opcagt -start
```

Después de configurar el agente para que se ejecute como usuario no raíz, el siguiente mensaje de error puede aparecer en el archivo **System.txt**:

```
ovbbccb (22461/1): (bbc-188) Cannot change the root directory for
the current process.
```

Ignore este error.

Cambio del usuario predeterminado para los 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 HPE 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 HPE 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 *<contraseña_cifrada>* por la salida del comando **opcpwcrpt** *<contraseña>*. Puede iniciar este comando desde un símbolo del sistema de comando en el servidor de administración.

  En Windows:

  Ejecute como `$OVO_STD_USER`

  En HP-UX/Linux/Solaris:

  `$OVO_STD_USER` seguido por la acción

Nota: La variable OVO_STD_USER está disponible ahora únicamente para la plataforma Windows.

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.

Nota: 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 contraseña.

Configuración del componente de seguridad para la clave asimétrica

Se usan los certificados RSA y el cifrado asimétrico para la comunicación segura durante el intercambio de señales de SSL entre los nodos y entre el nodo y el servidor de gestión.

Al instalar HPE Operations Agent:

1. El componente Certificate Client (OvSecCc), define la variable de configuración ASYMMETRIC_KEY_LENGTH en 2048 (valor predeterminado)
2. El componente Certificate Management (OvSecCm) crea un par de claves RSA de 2048 bits para la comunicación. Este par de claves RSA se usa para una comunicación segura.
3. Se crean los certificados CA con el alias <CA_ovcoreid_keylength>.

   Ejemplo: CA_8cd78962-ab51-755c-1279-85f5ba286e97_2048

   2048 es la longitud predeterminada de la clave.

Para incrementar la seguridad del cifrado, debe aumentar la longitud de la clave RSA. Use la variable de configuración ASYMMETRIC_KEY_LENGTH para definir la longitud de la clave RSA en el espacio de nombres `sec.cm`.

Siga los pasos para definir la variable de configuración ASYMMETRIC_KEY_ LENGTH y aplicar los cambios en la configuración en el servidor de administración y los nodos administrados:

En el servidor de administración

Siga estos pasos para actualizar la variable de configuración en el servidor de gestión:

1. Actualice la variable de configuración ASYMMETRIC_KEY_LENGTH usando el siguiente comando:

   `ovconfchg –ns sec.cm -set ASYMMETRIC_KEY_LENGTH` *<longitud de clave admitido por el algoritmo de cifrado>*
2. Vaya a la siguiente ubicación del servidor de gestión:

   En Windows:

   `%ovinstalldir%\lbin\seccs\install\`

   En HP-UX/Linux/Solaris:

   `%ovinstalldir%/lbin/seccs/install/`
3. Ejecute la herramienta MigrateAsymKey:

   En Windows:

   `cscript MigrateAsymKey.vbs -createCAcert`

   En HP-UX/Linux/Solaris:

   `./MigrateAsymKey.sh -createCAcert`

   MigrateAsymKey:
   - Crea un nuevo par de claves de CA.
   - Cea un certificado de CA en función de la longitud de la nueva clave definida en la variable de configuración ASYMMETRIC_KEY_LENGTH.
   - Actualiza los certificados de confianza para el agente local y todos los demás grupos de recursos OV (ovrgs) en el servidor de administración,
   - Crea un certificado para el agente local y todos los ovrgs.

   Nota: Durante una actualización del componente Certificate Server, la ejecución de la herramienta de migración con la opción -createCAcert se realiza después de la instalación. -createCAcert se usa para crear un certificado de CA correspondiente al nuevo valor de configuración establecido en la variable ASYMMETRIC_KEY_LENGTH y actualizar el certificado de CA creado por el agente en el servidor y todos los ovrg. Por tanto, no ejecute la herramienta de migración con la opción -createCAcert a no ser que se haya realizado un cambio de configuración en la variable de configuración ASYMMETRIC_KEY_LENGTH.

4. Para actualizar los certificados de confianza en todos los nodos administrados, ejecute el siguiente comando:

   ```
   ovcert –updatetrusted
   ```

   Nota: Debe ejecutar el comando *ovcert –updatetrusted* en todos los nodos gestionados antes de ejecutar la herramienta MigrateAsymKey con la opción --createNodecert.

5. Para crear un nuevo certificado de nodos para el agente en el servidor de gestión y todos los ovrg en el servidor, ejecute el comando:

   En Windows:

   ```
   cscript MigrateAsymKey.vbs -createNodecert
   ```

   En HP-UX/Linux/Solaris:

   ```
   ./MigrateAsymKey.sh -createNodecert
   ```

   El comando crea los nuevos certificados de nodo para Operations Agent en el servidor de gestión y todos los ovrg con el nuevo par de claves RSA:

Nota: Para ejecutar la longitud de las claves actualizadas, ejecute el comando: `ovcert –certinfo` <alias de certificado> La longitud de las claves se actualiza con ASYMMETRIC_KEY_LENGTH.

En el nodo del agente

Para aplicar los cambios de configuración en el nodo del agente (nodo gestionado), siga estos pasos:

1. Actualice la variable de configuración ASYMMETRIC_KEY_LENGTH usando el siguiente comando:

   ```
   ovconfchg –ns sec.cm -set ASYMMETRIC_KEY_LENGTH <longitud de clave admitido por el algoritmo de cifrado>
   ```

2. Para suprimir el certificado del nodo existente en el agente, ejecute los comandos siguientes:

   ```
   ovcert –remove<alias de certificado>
   ```

   ```
   ovcert –remove <alias de certificado CA>
   ```

3. Para solicitar un nuevoun certificado de nodo en el servidor de gestión, ejecute el comando:

   ```
   ovcert -certreq
   ```

Configuración del componente de seguridad para la clave simétrica

Cuando instala HPE Operations Agent, los componentes de seguridad OvSecore y OvJSecCore continuarán usando los valores del algoritmo predeterminado más antiguos para el cifrado y descifrado. Tanto OvSecCore como OvJSecCore se incluyen en los paquetes HPSharedComponent para Windows de la versión 11.10 en adelante.

Los algoritmos simétricos admitidos son los siguientes:

- Blowfish
- DES
- DES3
- AES128
- AES192
- AES256

Para cambiar el algoritmo de clave simétrica predeterminado, utilice las siguientes variables de configuración:

- **DEF_SYM_KEY_ALGO**- Esta variable se usa para establecer el algoritmo de clave simétrica predeterminado para el cifrado. Establezca esta variable de configuración en el espacio de nombres `sec.core`. Los valores admitidos para el algoritmo son los siguientes:
  - eBlowfish
  - eDES
  - eDES3
  - eAES128
  - eAES192
  - eAES256
  - eDefault – usa AES128 como algoritmo predeterminado.

Nota: Para habilitar el cumplimiento de FIPS, use uno de los siguientes algoritmos simétricos de cumplimiento de FIPS:

- 3DES
- AES128
- AES198
- AES256

- **ENABLE_DEF_SYM_KEY_ALGO**: defínala en TRUE para habilitar el uso del algoritmo de clave simétrica predeterminado establecido en DEF_SYM_KEY_ALGO.

  Nota: La herramienta MigrateSymKey establece la variable de configuración ENABLE_DEF_SYM_KEY_ALGO en TRUE, si aún no lo está.

Siga los pasos para actualizar la configuración en el servidor de administración y en los nodos que se están ejecutando con HPE Operations Agent:

1. Ejecute el siguiente comando para establecer la variable de configuración DEF_SYM_KEY_ALGO en cualquiera de los algoritmos admitidos:

   `ovconfchg –ns sec.core –set DEF_SYM_KEY_ALGO` *<Algoritmo compatible>*

   Nota: Si la variable DEF_SYM_KEY_ALGO no está establecida y la variable ENABLE_DEF_SYM_KEY_ALGO está establecida en TRUE, se utiliza eAES128 como algoritmo predeterminado.

2. Vaya a la siguiente ubicación en el servidor de gestión o en el nodo donde se instala el agente:

   En Windows:

   `%ovinstalldir%\bin\secco\`

   En HP-UX/Linux/Solaris:

   `%ovinstalldir%/lbin/secco/`

3. Ejecute la herramienta MigrateSymKey:

   En Windows:

   ```
   MigrateSymKey -sym_key_algo [eBlowfish | eDES | eDES3 |
   eAES128 | eAES192 | eAES256]
   ```

   En HP-UX/Linux/Solaris:

   ```
   ./MigrateSymKey.sh -sym_key_algo [eBlowfish | eDES | eDES3 |
   eAES128 | eAES192 | eAES256]
   ```

La herramienta MigrateSymKey establece la variable de configuración ENABLE_DEF_SYM_KEY_ALGO en TRUE y migra el contenido de KeyStore basado en el algoritmo establecido en la variable de configuración DEF_ SYM_KEY_ALGO.

Uso:

```
MigrateSymKey
        -sym_key_algo [eBlowfish | eDES | eDES3 | eAES128 | eAES192 | eAES256]
        -help
```

`-sym_key_algo`:

- Actualice "ENABLE_DEF_SYM_KEY_ALGO" en TRUE si no está establecido.
- Si se especifica el algoritmo de clave simétrica, DEF_SYM_KEY_ALGO se actualiza con el valor especificado.
- Si no se especifica el algoritmo, eAES128 se usa como algoritmo de clave simétrica.

Configuración del componente de seguridad con el algoritmo hash

HPE Operations Agent 12.01 y superiores admiten el algoritmo hash configurable para el uso de hash. **HASH_ALGO** es la variable de configuración proporcionada para establecer el algoritmo hash. La variable de configuración está disponible bajo el espacio de nombres **sec.core**.

HASH_ALGO admite los siguientes algoritmos hash:

- eMD5
- eSHA1
- eSHA224
- eSHA256
- eSHA384
- eSHA512
- eDefault

Nota: Para habilitar el cumplimiento de FIPS, use uno de los siguientes algoritmos hash de cumplimiento de FIPS:

- SHA1
- SHA224
- SHA256
- SHA384
- SHA512

Establezca la variable de configuración HASH_ALGO en cualquiera de los algoritmos admitidos mediante el comando siguiente:

`ovconfchg –ns sec.core –set HASH_ALGO` *<Algoritmo compatible>*

Ejemplo:`ovconfchg –ns sec.core –set HASH_ALGO=eSHA224`

Si la variable HASH_ALGO se establece en eDefault (o en cualquier otro valor no admitido), en este caso sec.core usa eSHA256 como valor de HASH_ALGO.

Hash también se usa para proteger el almacén de claves de HPE Operations Agent. Para habilitar el uso del algoritmo especificado en **HASH_ALGO** para proteger el almacén de clave, debe usar la herramienta MigrateSymKey. La herramienta MigrateSymKey establece la variable de configuración **HASH_ ALGO_AS_SEED** en TRUE en el espacio de nombres `sec.core` y migra el contenido del almacén de claves.

Nota:

- Ante de definir la variable de configuración HASH_ALGO_AS_SEED en TRUE, asegúrese de que la versión de los componentes OvSecCore y OvJSecCore es 12.00 o superior.
- Después de definir HASH_ALGO_AS_SEED en TRUE, solo debe usar la herramienta MigrateSymKey para actualizar el valor de HASH_ALGO. No utilice ovconfchg para definir HASH_ALGO_AS_SEED.
- El servidor de administración y el nodo de la gente deben usar el mismo valor de HASH_ALGO para que ovcert -certreq y ovcm -grant funcionen cuando HASH_ALGO_AS_SEED está definido como TRUE.
- No revierta el valor de HASH_ALGO_AS_SEED después de definirlo.

Siga los pasos para usar la herramienta MigrateSymKey:

1. Vaya a la siguiente ubicación en el servidor de administración o en el nodo en el que está instalado HPE Operations Agent:

   En Windows:

`%ovinstalldir%\lbin\secco\`

En HP-UX/Linux/Solaris:

`/opt/OV/lbin/secco/`

En AIX:

`/usr/lpp/OV/lbin/secco/`

2. Ejecute la herramienta de migración.

   `MigrateSymKey -hash_algo [eMD5 | eSHA1 | eSHA224 | eSHA256 | eSHA384 | eSHA512]`

Después de ejecutar la herramienta, el contenido de KeyStore se migra en función del nuevo valor especificado, y HASH_ALGO_AS_SEED se define como TRUE.

Uso:

```
MigrateSymKey
 -hash_algo [eMD5 | eSHA1 | eSHA224 | eSHA256 | eSHA384 | eSHA512]
 -help
```

- `-hash_algo` actualiza lo siguiente:
  - Actualiza "HASH_ALGO_AS_SEED" aTRUE, si no está establecido.
  - Actualiza la variable de configuración "HASH_ALGO" al algoritmo especificado.
  - Si no se especifica ningún algoritmo, se actualiza con eSHA256.

Verificación

- Si la herramienta MigrateSymKey completa la acción correctamente, imprime los siguientes mensajes: `"Migration successful“.`
- Después de ejecutar la herramienta MigrateSymKey, verifique si HASH_ALGO_AS_SEED está definido en TRUE y HASH_ALGO está definido en el algoritmo especificado.

Configuración de HPE Operations Agent para que cumpla los estándares FIPS

Los Estándares Federales de Procesamiento de la Información (FIPS, Federal información Processing Standard) son unos estándares de seguridad utilizados para endosar módulos criptográficos que se utilizan para proteger la información desclasificada, valiosa y confidencial en productos de hardware y software. Para

forzar una comunicación segura en un entorno de red y administrar todos los requisitos de los módulos criptográficos, HPE Operations Agent se ha mejorado para que cumpla los estándares FIPS.

Requisitos para que HPE Operations Agent cumpla los estándares FIPS

1. Asegúrese de que está instalado HPE Operations Agent version 12.00 o superior.
2. Asegúrese de que la longitud de la clave RSA sea superior o igual a 2048 bits.

   Para definir o cambiar la longitud de la clave RSA, consulte Configuración del componente de seguridad para la clave asimétrica.
3. Asegúrese de usar un algoritmo simétrico que cumpla el estándar FIPS para el cifrado y descifrado. De forma predeterminada, utiliza, AES128 como algoritmo simétrico que cumple el estándar FIPS. Para cambiar el algoritmo simétrico predeterminado, consulte Configuración del componente de seguridad para la clave asimétrica.
4. Asegúrese de usar un algoritmo hash que cumpla el estándar FIPS para el cifrado y descifrado. Para cambiar el algoritmo hash predeterminado, consulte Configuración del componente de seguridad con el algoritmo hash.

Habilitación del cumplimiento de FIPS con FIPS_tool

Use la herramienta **FIPS_tool** proporcionada por el componente OvSecCore para habilitar el cumplimiento de FIPS. La herramienta FIPS_tool está disponible en la siguiente ubicación:

En Windows

```
%ovinstalldir%\lbin\secco\
```

En Linux

```
/opt/OV/lbin/secco/
```

Uso:

```
FIPS_tool
        -enable_FIPS [-Java_Home <ruta_del_directorio_jre>]
        -disable_FIPS [-Java_Home <ruta_del_directorio_jre>]
        -help
```

Siga estos pasos para habilitar el modo FIPS con herramienta FIPS_tool:

1. Vaya a la siguiente ubicación en el servidor de administración o en el nodo en el que está instalado HPE Operations Agent:

   En Windows

   `%ovinstalldir%\bin\secco\`

   En UNIX:

   `/opt/OV/lbin/secco/`

   En AIX

   `/usr/lpp/OV/lbin/secco`
2. Ejecute la herramienta FIPS_tool para habilitar el modo FIPS:

   `FIPS_tool -enable_FIPS [-Java_Home <ruta_del_directorio_jre>]`

   La variable de configuración ENABLE_FIPS_MODE está establecida en TRUE.

   En esta instancia, `-Java_Home` especifica la ruta del directorio `jre`.

   Por ejemplo:

   ```
   FIPS_tool -enable_FIPS -Java_Home /opt/OV/nonOV/jre/b
   FIPS_tool -enable_FIPS -Java_Home /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.51-2.4.5.5.el7.x86_64/jre
   ```

   Nota: Ejecute la herramienta FIPS_tool para deshabilitar el modo FIPS:

   `FIPS_tool -disable_FIPS [-Java_Home <ruta_del_directorio_jre>]`

   Por ejemplo:

   ```
   FIPS_tool -disable_FIPS -Java_Home /opt/OV/nonOV/jre/b
   FIPS_tool -disable_FIPS -Java_Home /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.51-2.4.5.5.el7.x86_64/jre
   ```

   Si ejecuta la herramienta FIPS_tool para deshabilitar el modo FIPS, este se deshabilita, pero los cambios en la configuración realizados en el modo FIPS no se revierten.

Opciones de configuración en el modo FIPS

```
[sec.core]
DEF_SYM_KEY_ALGO=<algoritmo simétrico que cumple con FIPS>
ENABLE_DEF_SYM_KEY_ALGO=TRUE
ENABLE_FIPS_MODE=TRUE
HASH_ALGO=<algoritmo hash que cumple con FIPS>
```

```
HASH_ALGO_AS_SEED=TRUE
[sec.cm]
ASYMMETRIC_KEY_LENGTH=<tamaño de clave RSA que cumple con FIPS>
[eaagt]
CRYPT_USING_LCORE=TRUE
```

Por ejemplo:

```
[sec.core]
DEF_SYM_KEY_ALGO=eAES128
ENABLE_DEF_SYM_KEY_ALGO=TRUE
ENABLE_FIPS_MODE=TRUE
HASH_ALGO=eSHA256
HASH_ALGO_AS_SEED=TRUE
[sec.cm]
ASYMMETRIC_KEY_LENGTH=2048
[eaagt]
CRYPT_USING_LCORE=TRUE
```

Nota:

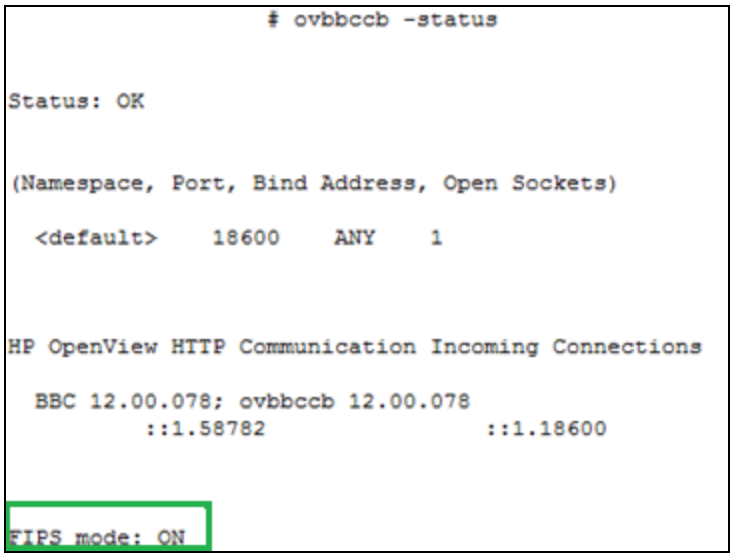
Si habilita el modo FIPS en un nodo de HPE Operations Agent, debe habilitar el modo FIPS en el servidor de administración y, después, volver a implementar todas las directivas desde el servidor de administración.

Verificación de si HPE Operations Agent se está ejecutando en el modo FIPS

Ejecute el comando siguiente:

```
ovbbccb –status
```

Si HPE Operations Agent se está ejecutando en el modo FIPS, el modo FIPS en la salida del comando es ON.



Solución de problemas

La comunicación SSL entre el servidor de Java y el cliente de Java falla en el modo FIPS.

La comunicación SSL entre el servidor de Java y el cliente de Java falla en el modo FIPS porque el campo Comentario de Netscape se trata como el campo TipoCert de Netscape.

Para resolver este problema, el campo Comentario de Netscape (HP OpenView Certificate) se elimina de OvSecCm en la versión actual (12.01) de HPE Operations Agent y los certificados se vuelven a emitir.

Nota: Ejecute el siguiente comando para comprobar los detalles del certificado:

openssl x509 –in <cert.pem> -text –noout

Mensaje de SSL no reconocido, conexión de texto sin formato.

El problema se produce debido a un retardo causado por los jar de RSA Bsafe durante el intercambio de señales de SSL cuando FIPS está habilitado.

Para resolver el problema, establezca la siguiente configuración en el servidor de administración para aumentar el tiempo de espera del intercambio de señales de SSL:

```
[bbc.http]

SSL_HANDSHAKE_TIMEOUT=60000
```

Excepción de E/S: error de conexión SSL (sec.core-114) (no se pueden encontrar los parámetros ecdh).

Éste problema solo se produce cuando FIPS está habilitado. El algoritmo ECDHE no se admite en el modo FIPS. Este error aparecerá después de habilitar FIPS si el algoritmo ECDHE se toma por defecto durante la comunicación SSL.

Para resolver este problema, deshabilite el algoritmo ECDHE en el archivo `java.security`:

```
jdk.tls.disabledAlgorithms= ECDHE
```

Monitorización de los registros de eventos de Windows

El componente Logfile Encapsulator de HPE Operations Agent admite la función de reenvío de eventos del registro de eventos de Windows. Es decir, se pueden leer los eventos que se reenvían desde distintas máquinas y también puede monitorizar los registro de eventos de Windows. Las directivas del registro de eventos de Windows le ayudan a configurar el agente para monitorizar los registros de eventos de Windows que prefiera.

Las siguientes versiones de Windows proporcionan una nueva categoría de registros de eventos (registros de aplicaciones y servicios):

- Windows Vista
- Windows Server 2008
- Windows Server 2008 R2
- Windows 7

Estos registros de aplicaciones y servicios se pueden monitorizar con HPE Operations Agent con directivas del registro de eventos de Windows correctamente configuradas.

HPE Operations Agent no puede monitorizar los siguientes tipos de registros de eventos:

- Los eventos que se originan en un sistema remoto (recopilados con la función de suscripción de eventos de Windows)
- Registros de eventos guardados.

HPE Operations Agent puede monitorizar los eventos con los siguientes niveles de eventos:

- Error
- Information
- Advertencia
- LOG_ALWAYS
- VERBOSE
- Audit Failure
- Audit Success

La tabla siguiente muestra cómo se visualizan los campos del registro de eventos en el explorador de mensajes.

Tabla 1. Correlación de los campos de registro de eventos con el explorador de mensajes

Campo del registro de eventos	Campo del explorador de mensajes	Comentarios
Date	Date	La fecha de creación del evento en el nodo gestionado.
Time	Time	La hora de creación del evento en el nodo gestionado.
Event ID	Message Text	El ID de evento se mostrará antes de cualquier texto del mensaje adicional.
Source	Application	Ninguno
Type *error* *error* *information* *warning* *log_always* *verbose*	Severity *critical* *error* *normal* *warning* *normal* *normal*	La asignación de la gravedad del tipo de registro de eventos a la gravedad de mensajes de Operations Manager.

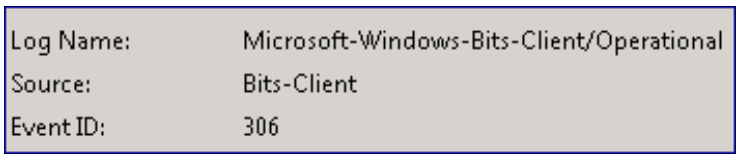
Tabla 1. Correlación de los campos de registro de eventos con el explorador de mensajes , continuación

Campo del registro de eventos	Campo del explorador de mensajes	Comentarios
audit failure *audit success*	*warning* *normal*	
Category	Object	Ninguno
Description	Message Text	Todos los demás textos de mensaje (después del ID de evento).
User	Not mapped	No asignado
Computer	Node	Obtenga el nombre del nodo que aparece en el servidor de gestión.
	Msg Group	Vacío

Supervisión de registros de eventos de aplicaciones y servicios desde HPOM para Windows

Para crear una directiva de registro de eventos de Windows para monitorizar un registro de aplicaciones y servicios, siga estos pasos:

1. Inicie sesión en el nodo de Windows en que se encuentre el registro de eventos de Windows.
2. Abra la ventana del visor de eventos.
3. En el árbol de la consola, seleccione el evento. En el panel de detalles, aparece el nombre del registro de eventos (junto al campo Log Name).

   Anote el nombre del archivo de registro tal como aparece en el panel de detalles.
4. Abra la consola de HPOM para Windows.

5. En el árbol de la consola, en Agent Policies Grouped by Type, haga clic con el botón secundario en **Windows Event Log** y, a continuación, haga clic en **New > Policy**.

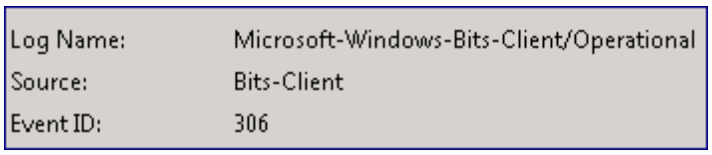
   Se abre el editor de directivas de la directiva Windows Event Log .
6. En la ficha Source, escriba el nombre del registro de eventos de Windows (que anotó en el paso 3) en el campo Event Log Name.

7. Para especificar los restantes detalles de la directiva, siga las instrucciones de la ayuda en línea de HPOM para Windows.
8. Guarde la directiva.
9. Implemente la directiva en el nodo de Windows.

Supervisión de registros de eventos de aplicaciones y servicios desde HPOM en UNIX/Linux 9.xx

Para crear una directiva de registro de eventos de Windows para monitorizar un registro de aplicaciones y servicios, siga estos pasos:

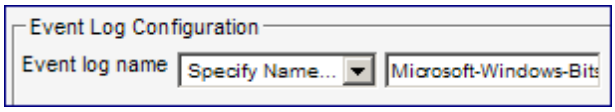
1. Inicie sesión en el nodo de Windows en que se encuentre el registro de eventos de Windows.
2. Abra la ventana del visor de eventos.
3. En el árbol de la consola, seleccione el evento. En el panel de detalles, aparece el nombre del registro de eventos (junto al campo Log Name)



   Anote el nombre del archivo de registro tal como aparece en el panel de detalles.
4. Inicie una sesión en la interfaz de usuario de administración de HPOM para UNIX.
5. Haga clic en **OMU**.
6. Haga clic en **Browse > All Policy Types**.
7. Haga clic en el registro de eventos de Windows. Se abre la página **Windows_ Event_Log** del tipo de directiva.

8. Haga clic en  y, a continuación, en **New Policy**. Se abre la página Add Windows_Event_Log Policy.

   En la ficha Source, en el campo Event Log Name, seleccione Specify Name. Aparece un nuevo cuadro de texto, en el que debe escribir el nombre del registro de eventos de Windows (que anotó en el paso 3).

9. Para especificar los restantes detalles de la directiva, siga las instrucciones de la ayuda en línea de HPOM para UNIX.
10. Guarde la directiva.
11. Implemente la directiva en el nodo de Windows.

Capítulo 2: 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.

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 *GlancePlus Online Help*.

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.

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**, HPE 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 HP-UX/Linux/Solaris:

```
/var/opt/perf/perfd
```

Para ejecutar una secuencia de comandos que incluya diagnosis y acciones específicas del sistema operativo, deberán usarse las secuencias predeterminadas siguientes:

En Windows:

```
%ovdatadir%\perf\perfd\os\<tipo_de_so>\adv
```

En HP-UX/Linux/Solaris:

```
/var/opt/perf/perfd/os/<tipo_de_so>/adv
```

En este caso, *<tipo_de_so>* especifica 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 HP-UX/Linux/Solaris:

`/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` *<secuencia_de_comandos>*`-n`*<secuencia_de_comandos>*

   La secuencia de asesor comienza a ejecutarse en el sistema especificado y genera resultados basados en la configuración del archivo de secuencia.

   Sugerencia: Al usar los scripts en un sistema remoto,, asegúrese de que el proceso `perfd` se ejecuta en el sistema remoto. Puede impedir que otros sistemas ejecuten el script del asesor en el sistema local. Para más información, consulte Restricción de acceso.

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 HP-UX/Linux/Solaris:

   `/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>*

El script 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 script.

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\perfd`

En HP-UX/Linux/Solaris:

`/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 item opcional.
- Items separados por comas con corchetes o llaves son opciones. Seleccione sólo uno.
- 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

/ / *[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]
             ("==" means "equal", and "!=" means "not equal".)
```

Las condiciones son utilizadas en las instrucciones ALARM e 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: entero y real. 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 bydsk_)
- 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 percentajes
- 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 = ""
```

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.

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.

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
```

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).

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.

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 una 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.

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 finaliza la 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.

Instrucción PRINT

Utilice la instrucción PRINT para imprimir los datos que está recopilando en la salida estándar (la consola de comandos `padv`). 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, "%"
```

Cuando se inicia, esta instrucción imprime un mensaje en la consola de comandos `padv` como el siguiente:

Nota: 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` se refiere al 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.

Capítulo 3: Alarmas de rendimiento

Componente Performance Collection permite al usuario definir alarmas. Estas alarmas notifican al usuario si **oacore** o la métrica DSI 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 los datos registrados por **oacore** u otros recopiladores, se compara con las definiciones de alarma en el archivo `alarmdef`. Si **oacore** o las métricas DSI cumplen o exceden las condiciones definidas, se desencadenará una alerta o acción.

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 con las definiciones de alarma y notificar los resultados con el comando `analyze` del programa `utility`.

Para obtener más información sobre la definición de alarmas para las métricas DSI, consulte Definición de alarmas para las métricas DSI en el capítulo *Uso de datos DSI registrados en el Almacén de datos de métricas.*

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).

No obstante, si los datos no están registrados en los archivos de base de datos (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`. Para obtener más información, consulte Umbrales.

Entre las acciones definidas en la definición de alarma se encuentran:

- acciones locales realizadas con los comandos del sistema operativo
- mensajes enviados a Network Node Manager (NNM) y a HPOM

Generador de alarmas

El componente generador de alarmas del Componente Performance Collection procesa el archivo `alarmdef` y los datos de rendimiento del sistema disponibles y, a continuación, genera alarmas en caso necesario. El generador de alarmas consta de los componentes siguientes:

- Servidor del generador de alarmas (`perfalarm`)
- Base de datos del generador de alarmas (`agdb`)

El servidor del generador de alarmas explora la información contenida en el archivo `alarmdef` y envía alertas a los destinos en función de la información de configuración del archivo `alarmdef`. La base de datos `agdb` contiene una lista de sistemas de destino para que el componente `perfalarm` reenvíe capturas SNMP ante eventos específicos. El usuario puede modificar el comportamiento predeterminado del componente `perfalarm` y acceder a los datos disponibles en la base de datos `agdb` con ayuda de la utilidad `agsysdb`.

Ejecute el comando siguiente para ver la lista de sistemas de destino donde se envían las notificaciones de alerta.

```
agsysdb -l
```

Habilitación perfalarm

De manera predeterminada, el servidor de generador de alarmas (`perfalarm`) está deshabilitado. Use uno de los métodos siguientes para habilitar `perfalarm`:

- **Antes de instalar HPE Operations Agent:**
  a. Establezca la variable ENABLE_PERFALARM en **True** en el archivo de perfil:

     ```
     set nonXPL.config:ENABLE_PERFALARM=TRUE
     ```

  b. Ejecute el comando siguiente para instalar HPE Operations Agent con el archivo de perfil:

     En Windows:

     `cscript oainstall.vbs -i -a -agent_profile` *<ruta>*`\`*<archivo_de_perfil>* `-s` *<dirección_IP_del_servidor>*

     En HP-UX/Linux/Solaris:

`./oainstall.sh -i -a -agent_profile` *<ruta>*`/`*<archivo_de_perfil>* `-s` *<dirección_IP_del_servidor>*

En este ejemplo:

<archivo_de_perfil> es el nombre del archivo de perfil.

<ruta> es la ruta completa de acceso al archivo de perfil.

<dirección_ip_servidor> es la dirección IP o nombre de host del servidor de gestión.

- **Después de instalar HPE Operations Agent:**
  a. Ejecute los comandos siguientes para habilitar `perfalarm`:

  En Windows:

  ```
  copy "%ovinstalldir%newconfig\alarmdef.mwc" "%ovdatadir%"
  "%ovinstalldir%bin\ovpacmd" start alarm
  ```

  En HP-UX/Linux/Solaris:

  ```
  cp/opt/perf/newconfig/alarmdef /var/opt/perf/
  /opt/perf/bin/ovpa start alarm
  ```

  En AIX:

  ```
  cp/usr/lpp/perf/newconfig/alarmdef /var/opt/perf/
  /usr/lpp/perf/bin/ovpa start alarm
  ```

  b. Vaya a la siguiente ubicación en el nodo:

  En Windows 64-bit:

  `%ovinstalldir%bin\win64\OpC\install`

  En otros Windows:

  `%ovinstalldir%bin\OpC\install`

  En HP-UX/Linux/Solaris:

  `/opt/OV/bin/OpC/install`

  En AIX:

  `/usr/lpp/OV/bin/OpC/install`

  c. Ejecute el comando siguiente para volver a configurar HPE Operations Agent:

  En Windows:

  `cscript oainstall.vbs -a -configure -agent_profile`*<ruta>*`/`*<archivo_de_perfil>*

En HP-UX/Linux/Solaris:

`./oainstall.sh -a -configure -agent_profile`*<ruta>*`/`*<archivo_de_perfil>*

En este ejemplo:

<archivo_de_perfil> es el nombre del archivo de perfil.

<ruta> es la ruta completa de acceso al archivo de perfil.

Nota: Si actualiza desde HPE Operations Agent 11.xx en 12.00, perfalarm continúa funcionando de la manera definida anteriormente.

Envío de capturas SNMP a Network Node Manager

Para enviar capturas de SNMP al gestor de nodos de red, deberá agregar el nombre de su sistema en `agdb` de Componente Performance Collection con el comando siguiente:

`agsysdb -add` *nombre del sistema*

Cuando se genera una ALERTA, se enviará 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` *nombredelsistema*

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 snmp-trap. Si no está disponible esta entrada, las capturas no serán enviadas al otro nodo.

Envío de mensajes a HPOM

Por defecto, el generador de alarmas *no* ejecuta acciones locales que se hayan definido en las alarmas de la instrucción EXEC. En cambio, enviará un mensaje al explorador de eventos de HPOM.

Con el siguiente comando puede cambiar el valor predeterminado para dejar de enviar información a HPOM:

```
agsysdb -ovo OFF
```

Tabla 9: Configuración para enviar información a HPOM y ejecutar acciones locales

Indicador de HPOM	Componente Operations Monitoring En ejecución	Componente Operations Monitoring No ejecutándose
`off`	No se envían notificaciones de alerta a HPOM.	No se envían notificaciones de alerta a HPOM.
`on`	Se envían notificaciones de alerta a HPOM.	No se envían notificaciones de alerta a HPOM.

Ejecución de acciones locales

De manera predeterminada, Componente Performance Collection no ejecuta los comandos locales especificados en las instrucciones EXEC.

El valor predeterminado para deshabilitar las acciones locales se puede cambiar como se indica a continuación:

```
agsysdb -actions always
```

La siguiente tabla incluye las configuraciones para enviar información a HP Operations Manager (HPOM) y ejecutar acciones locales:

Tabla 10: Configuración para enviar información a HPOM y ejecutar acciones locales

Indicador de acciones locales	Componente Operations Monitoring En ejecución	Componente Operations Monitoring No ejecutándose
`off`	Sin acciones locales	Sin acciones

Tabla 10: Configuración para enviar información a HPOM y ejecutar acciones locales, continuación

Indicador de acciones locales	Componente Operations Monitoring En ejecución	Componente Operations Monitoring No ejecutándose
	ejecutadas.	locales ejecutadas.
`always`	Las acciones locales ejecutadas aunque se esté ejecutando Componente Operations Monitoring.	Acciones locales ejecutadas.
`on`	Acciones locales enviadas a HPOM.	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 -l
```

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` se puede usar para buscar condiciones de alarma en los datos del archivo de registro (consulte el capítulo 6, Comandos de utility). Esto difiere del procesamiento de alarmas en tiempo real explicado anteriormente ya que en este caso se comparan datos históricos del almacén de datos 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. Se enunciará un evento de repetición 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.

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
ADVERTENCIA: 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: 06/05/1999 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:

- ALERT: envía un mensaje a HPOM o una captura de SNMP a NNM
- EXEC: ejecuta un comando del sistema operativo
- PRINT: 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 más información, consulte 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 de sintaxis

- Las llaves de apertura y cierre ({ }) indican que una de las opciones es requerida.
- Los corchetes ([ ]) indican un item opcional.
- Items separados por comas con 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

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 *Componente Performance Collection Dictionary of Operating Systems Performance Metrics*.

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`.

Nota: 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.

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
```

Nota: 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:

- El espacio swap global ha estado casi lleno durante 5 minutos.
- La frecuencia de paginación de memoria ha sido demasiado alta durante 1 intervalo.
- El uso del CPU en ejecución es 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]
```

En este ejemplo:

- 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, podrá vincular las alarmas de Componente Performance Collection con los servicios que haya definido en ServiceNavigator. Para más información, consulte la guía *HP Operations ServiceNavigator Concepts and Configuration Guide*.

  ```
  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` es la acción que se usa más frecuentemente con ALARM START, REPEAT o END con 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` se define 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
```

Una alarma se genera cuando ambas condiciones son **True**.

`FOR` especifica la duración del tiempo en `SECONDS` o `MINUTES` durante el cual la condición debe permanecer en **True** para desencadenar una alarma.

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.

No fije una duración que sea menor que el intervalo de recopilación. Dado que el valor de la métrica no cambia hasta que finalice el siguiente ciclo de recopilación, el establecimiento de una duración inferior al intervalo de recopilación significa un procesamiento innecesario para generar alarmas duplicadas.

En el archivo `parm`, los siguientes parámetros del parámetro `collectioninterval` controlan el intervalo de recopilación:

- `global`: Este parámetro indica el intervalo de recopilación (en segundos) para todas las métricas que se registran en los archivos de registro oacore y `dsi`, excepto las métricas del proceso. De manera predeterminada, está establecido en cinco minutos.
- `process`: Este parámetro indica que el intervalo de recopilación (en segundos) para las métricas del proceso.

  Para más información acerca del parámetro `collectioninterval`, consulte Configuración de intervalos de registro de datos.
- `REPEAT EVERY`*n*`SECONDS, MINUTES` especifica el periodo de tiempo que transcurre para que se repita la alarma.

  En este ejemplo, **n** se refiere a la duración. Por ejemplo, `REPEAT EVERY`*5*`SECONDS, MINUTES`

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 *action* 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 HPOM y enviará una captura de SNMP a NNM.
Instrucción VAR

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"
```

Este ejemplo ALARM comprueba la métrica `gbl_swap_space_util` es superior a 90. En función de la configuración del generador de alarmas, ALERT se puede enviar a NNM a través de una captura de SNMP o como mensaje a Operations Manager.

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 enviar 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 desencadenar que se envíe una captura de SNMP a NNM o que se envíe un mensaje a HPOM. EXEC puede desencadenar que un mensaje de correo se envíe como acción local en el nodo, en función de la forma en que se haya configurado el generador de alarmas.

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 la métrica `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 la alerta RESET.

```
ALARM gbl_cpu_total_util > 85
  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 > 85
  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 gbl_cpu_total_util > 85 THEN    PROCESS LOOP...
```

Los diferentes intervalos no pueden ser sincronizados como se espera, de modo que los resultados no serán válidos.

Nota: 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 u 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 en lugar de esperar a 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 message`*

- `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 también puede desencadenar que una captura de SNMP se envíe a NNM o que un mensaje se envíe a HPOM, en función del modo en que haya configurado el generador de alarmas. En el caso de mensajes de alerta enviados a HPOM, las ADVERTENCIAS se mostrarán 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` *"comando del sistema"*

comando del sistema — 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 HPOM, normalmente no se realizarán las acciones locales.

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"
```

Nota: 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.

Para obtener más información, consulte "Instrucción PRINT" en la página 105.

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 *condición* **THEN** *acción* **[ELSE** *acción***]**

Condición – Una condición puede definirse como una comparición 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 !=.

acción – Cualquier acción o establecer variable (ALARM no es válido en este caso).

Modo en que es utilizado

La instrucción IF comprueba la *condición*. Si la *condición* es true, la *acción* después de THEN será ejecutada. Si la *condición* es false, entonces la *acción* dependerá de la cláusula opcional ELSE. Si se ha especificado la cláusula ELSE, la *acción* 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 NNM 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 gbl_cpu_total_util > 85 THEN PROCESS LOOP ...
```

Los diferentes intervalos no pueden ser 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 de varias instancias.
- *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 `alarmdef`:

```
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.

Nota: 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 más información sobre la sintaxis de nombres de métrica consulte la sección Nombre de métricas en este mismo 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.

`perfalarm` 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]` *nombre* `=` *valor*

name - Las variables deben empezar por una letra e incluir letras, dígitos y el caracter 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 caracter 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.

Para obtener más información, consulte "Instrucción SYMPTOM " en la página 105.

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 NNM. 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 NNM o el envío de un mensaje a HPOM 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:bydsk_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`

   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.

Capítulo 4: HPE Operations Agent en un entorno seguro

HPOM, junto con HPE Operations Agent, permite monitorizar y administrar sistemas y aplicaciones implementadas en un entorno de redes a partir de una consola central. En un entorno de administración basado en HPOM, el usuario puede comenzar a monitorizar los sistemas deseados tras instalar en ellos HPE Operations Agent. Con ayuda de las directivas implementadas de la consola de HPOM al nodo del agente, el usuario podrá habilitar diferentes funcionalidades de monitorización del agente.

Las responsabilidades principales del HPE Operations Agent en un entorno distribuido son:

- **Monitorizar datos**
  HPE Operations Agent puede comparar el valor de una métrica específica con un valor preestablecido y actuar consecuentemente en función de su configuración. Las directivas, implementadas de la consola de HPOM al nodo, desempeñan un papel importante como facilitadoras de la funcionalidad de monitorización de HPE Operations Agent.
- **Recopilar y almacenar datos**
  El recopilador de datos de rendimiento de HPE Operations Agent puede ser programado para recopilar y registrar los datos deseados en el sistema de monitorización. Se pueden agregar funcionalidades de recopilación adicionales instalando SPI y archivando los datos recopilados por los SPI en el almacén de datos del agente.

Directivas

Para trabajar con el agente se deberán implementar las recopilaciones de los detalles de configuración y especificaciones llamadas directivas en los nodos de administración de la consola de HPOM. En función del tipo de directiva implementada, se habilitaran determinados componentes de HPE Operations Agent. Las directivas proporcionan al agente la siguiente información:

- **Detalles de origen de monitorización**
  - Objetos a monitorizar.
  - Intervalo de sondeo para la monitorización de un objeto.
  - Valor de umbral del objeto monitorizado.

  - Reglas y condiciones para el análisis de datos en función del valor de umbral establecido.

- **Detalles del evento**
  HPE Operations Agent puede ser configurado por medio de directivas para generar eventos con mensajes, instrucciones e indicadores de gravedad cuando un objeto monitorizado infrinja la regla de umbral. Los eventos son reenviados a la consola de HPOM en forma de mensajes. El agente puede ser configurado de manera que realice ciertas acciones en respuesta a estos eventos.
- **Detalles de recopilación de datos**
  Si se desean monitorizar los datos recopilados por un programa externo, se podrá programar HPE Operations Agent para registrar los datos en su almacén de datos.

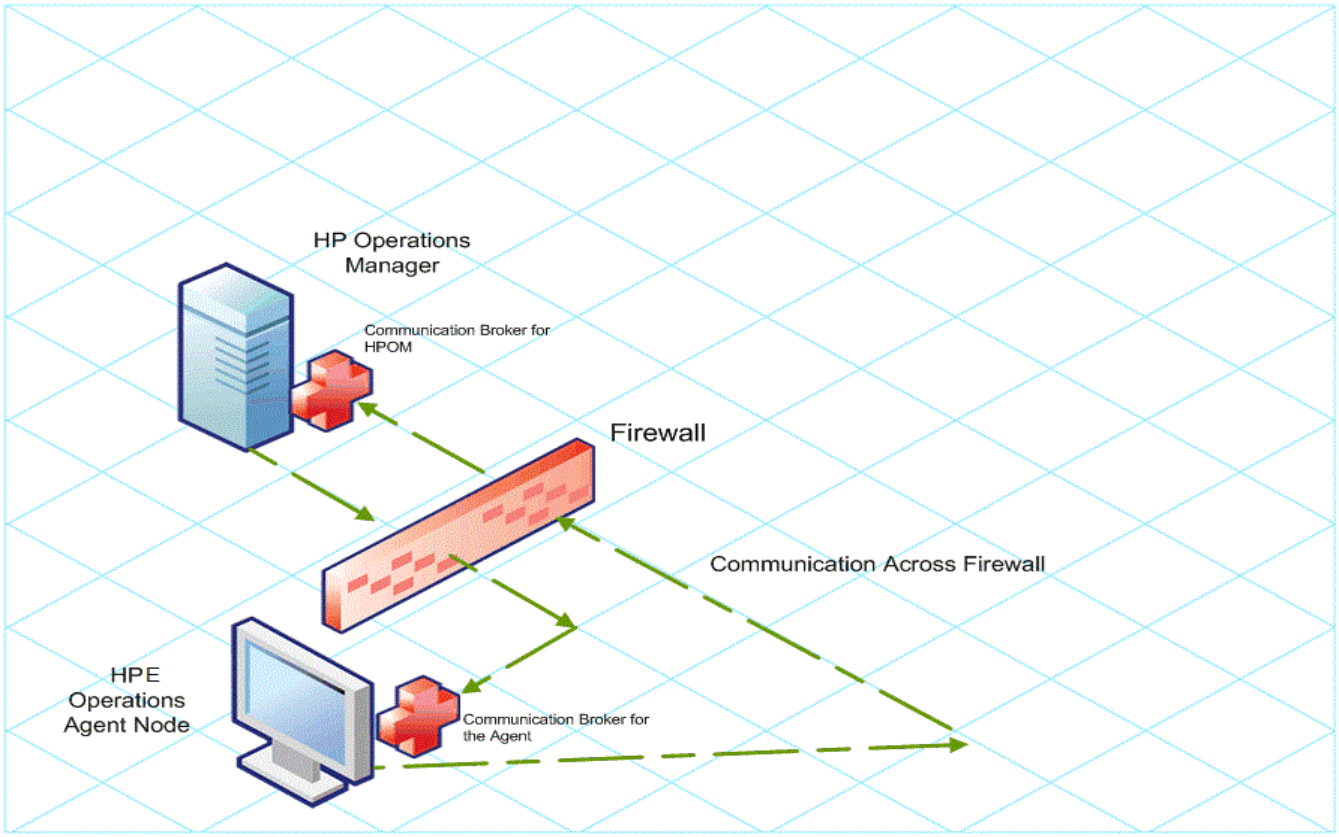
Modo de comunicación HTTPS

El modo de comunicación HTTPS permite a los nodos de HPE Operations Agent comunicarse fácilmente tanto entre sí como con otros productos estándares.

Ventajas de la comunicación HTTPS

- **Comunicación por cortafuegos**
  Con ayuda del protocolo HTTPS, los nodos de HPE Operations Agent pueden comunicarse con otros sistemas disponibles por cortafuegos. HPE Operations Agent puede ser implementado en un entorno seguro con cortafuegos y proxys.

  La figura siguiente muestra cómo traspasar un cortafuegos con la comunicación HTTPS.

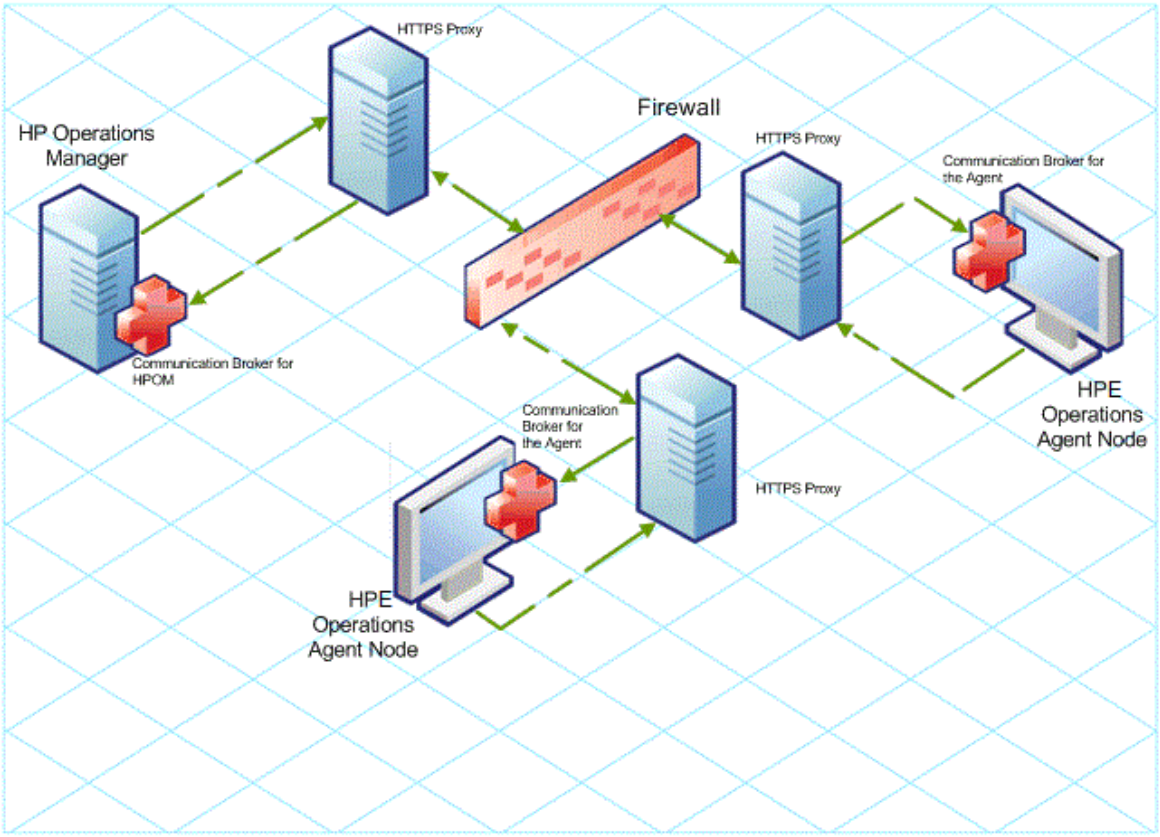


- **Seguridad avanzada**
  HPE Operations Agent usa el protocolo Secure Socket Layer (SSL) para restringir y controlar el acceso de usuario. Con ayuda del protocolo SSL, HPE Operations Agent comprime y cifra los datos involucrados en la comunicación de éste con otros sistemas.

  Asimismo, todos los mensajes remotos pasan por el agente de comunicación, con lo cual existe una única entrada de puerto al nodo de HPE Operations Agent.

  En un nodo de HPE Operations Agent, si se desean enviar mensajes, archivos u objetos, pueden configurarse uno o más proxys HTTP estándares para traspasar el cortafuegos o contactar un sistema remoto.

  La figura siguiente muestra cómo traspasar un cortafuegos con los proxys HTTPS externos.



- **Estándares abiertos**
  La comunicación HTTPS de HPE Operations Agent se basa en el protocolo estándar de la industria HTTP 1.1 y los sockets SSL. El cumplimiento de HPE Operations Agent de estándares abiertos tales como HTTP, SSL y SOAP, permite al usuario maximizar el uso de la actual infraestructura HTTP.
- **Escalabilidad**
  La comunicación HTTPS de HPE Operations Agent ha sido diseñada para funcionar correctamente, independientemente del tamaño del entorno y de la cantidad de datos enviados y recibidos. La comunicación HTTPS de HPE Operations Agent puede ser configurada según los requisitos de la organización.

Agente de comunicación

El componente Agente de comunicación ofrece una solución de puerto único para el nodo de HPE Operations Agent. Durante una implementación estándar, múltiples servidores pueden ser registrados con el nodo de HPE Operations Agent para la comunicación de datos. HPE Operations Agent envía las solicitudes de todos los servidores registrados en el nodo por medio del agente de

comunicación. El agente de comunicación reenvía de forma transparente la solicitud al servidor registrado de la misma manera que un proxy HTTP reenvía una solicitud HTTP. El puerto predeterminado del agente de comunicación es el 383. Es posible configurar HPE Operations Agent de manera que use un puerto diferente para el agente de comunicación.

Por razones de máxima seguridad, en los sistemas UNIX se inicia el agente de comunicación con `chroot`. `chroot` restringe la parte del sistema de archivos visible para el proceso del agente de comunicación obligando que una ruta específica actúe como directorio raíz y, por lo tanto, reduciendo al mínimo la exposición de acceso no autorizado.

El agente de comunicación se ejecuta como demonio en los sistemas UNIX y como servicio en los sistemas Windows.

El agente de comunicación usa como mínimo un puerto para aceptar datos de entrada en un nodo. El puerto está asociado con un identificador de nodo único (`OVCoreID`) para identificar el nodo. El usuario puede configurar el agente de comunicación de manera que se usen múltiples puertos con nodos de alta disponibilidad.

Escenarios de cortafuegos

Los cortafuegos pueden proteger de ataques externos a sistemas en una red. Los cortafuegos separan normalmente Internet de una Intranet privada. El usuario puede implementar múltiples niveles de cortafuegos para delimitar el acceso a entornos de confianza y restringir los de escasa confianza.

Los cortafuegos separan los entornos de red en dos zonas principales: la **zona de confianza** y la **zona desmilitarizada** (**DMZ**) (por ejemplo, Internet). La configuración del cortafuegos asegurará que la transmisión de datos de la DMZ a la zona de confianza esté restringida o controlada. En función de la configuración, el cortafuegos permitirá una **comunicación bidireccional** o una **comunicación de salida**.

Si se configura el cortafuegos en un entorno que permita una comunicación de tipo bidireccional, la red permitirá la comunicación HTTPS por el cortafuegos en ambas direcciones con algunas restricciones. El usuario podrá configurar los parámetros del cortafuegos para usar las opciones de configuración siguientes:

- *Proxys*: Si la red sólo permite la conexión de únicamente ciertos sistemas proxy por el cortafuegos, el usuario podrá redireccionar la comunicación de HPE Operations Agent por estos proxy.

- *Puertos locales*: Si la red sólo permite las conexiones de salida de ciertos puertos locales, se puede configurar HPE Operations Agent de manera que use puertos locales específicos.
- *Puertos del agente de comunicación*: Si la red sólo permite conexiones de entrada a ciertos puertos de destino distintos al puerto 383, se pueden configurar puertos de agentes de comunicación alternativos.

Si el cortafuegos del entorno sólo permite la comunicación de salida, se podrá configurar un **Reverse Channel Proxy** (**RCP**) con HPE Operations Agent. El RCP configurado con el nodo de HPE Operations Agent funciona como un proxy HTTP y permite al usuario transferir datos de una DMZ a una zona de confianza (segura). En lugar de comunicarse directamente con los sistemas HP Software, los RCP establecen un canal de comunicación con el agente de comunicación. El agente de comunicación comprueba y autentica la información que se origina en la zona DMZ y, a continuación, transfiere la información validada al nodo de HPE Operations Agent presente en la zona de confianza (segura).

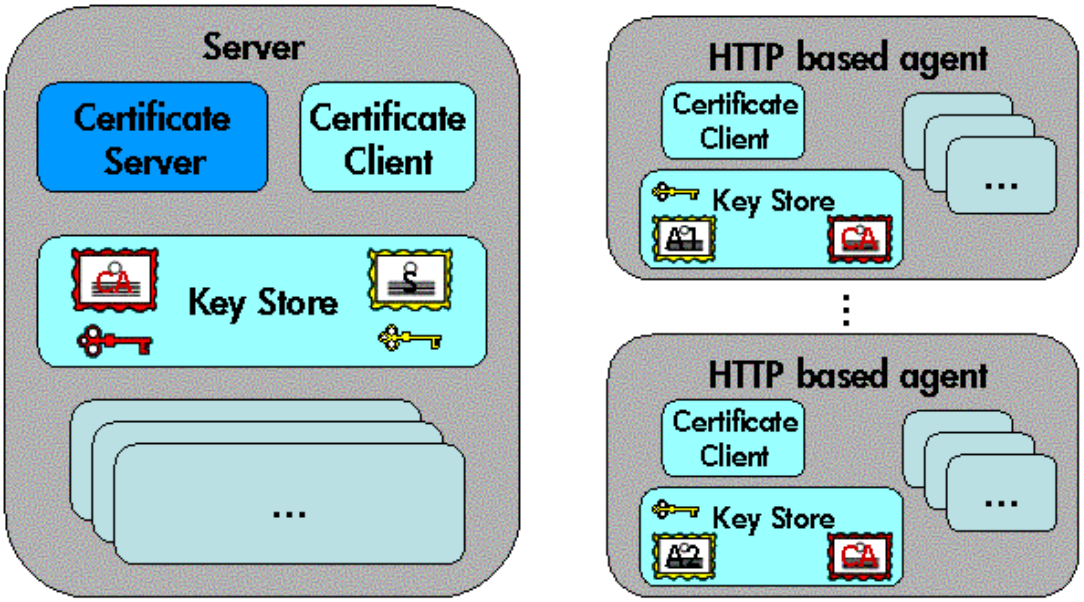
Componentes de seguridad basados en HTTPS

Para establecer comunicación con otros nodos de HPE Operations Agent o con el servidor de HPOM, el nodo de HPE Operations Agent deberá tener un certificado X509 estándar de la industria válido. Los nodos se comunican entre sí tras intercambiar certificados firmados por claves de 1024 bits. El intercambio de certificados permite que un nodo se identifique ante otro nodo o servidor del entorno administrado.

Los principales componentes responsables de crear y administrar certificados son:

- Servidor de certificados (reside en el servidor de HPOM)
- HPE Operations Agent Key Store
- HPE Operations Agent Certificate Client

La figura siguiente muestra estos componentes:

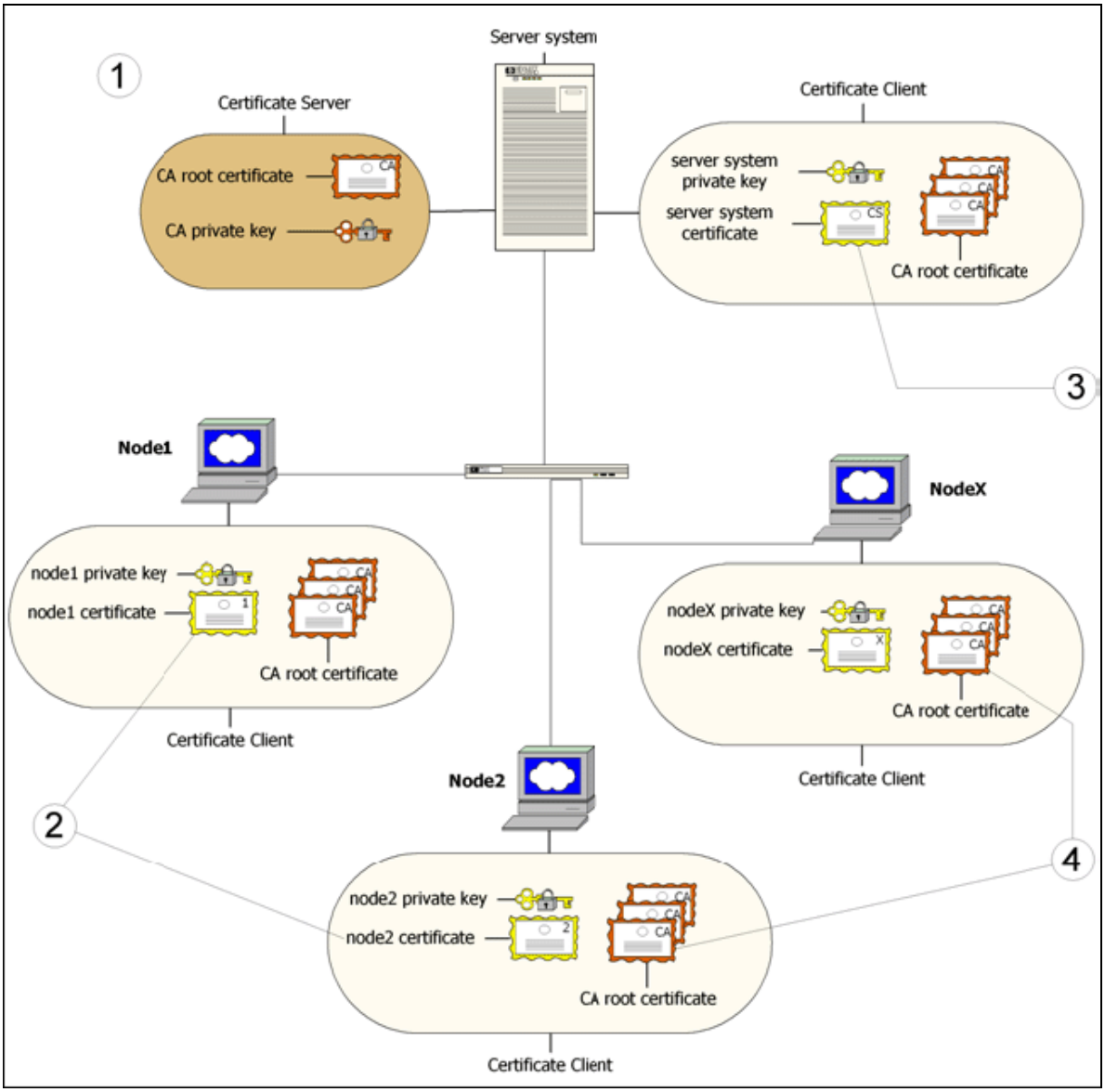


A cada sistema que hospeda un HPE Operations Agent se le asigna un valor identificador único para el parámetro `OvCoreId`, el cual es creado durante la instalación de HPE Operations Agent en dicho sistema.

Nota: El parámetro `OvCoreId` del nodo del agente permanece invariable, incluso si se producen cambios en el nombre del host o en la dirección IP del sistema.

Para cada nodo del agente, `OvCoreId` actúa como identificador único y está contenido en el certificado de nodo correspondiente. A `OvCoreId` se le asigna su valor durante la instalación.

La figura siguiente muestra un entorno de comunicación autenticada en una implementación de HPE Operations Agent.



1. Un sistema de servidor hospeda Certificate Server, el cual contiene la funcionalidad de autoridad de certificación (CA) requerida.
2. Cada sistema contiene un certificado asignado por Certificate Server con la clave privada de la autoridad de certificación.
3. El sistema de servidor requiere también un certificado para administrar su identidad.
4. Cada sistema contiene una lista de certificados raíz de confianza, los cuales deben contener al menos un certificado. Los certificados raíz de confianza (CA) permiten verificar la identidad de los asociados de comunicación. Un asociado de comunicación es de confianza si el certificado presentado puede ser validado mediante la lista de certificados de confianza.

Se requerirá una lista de certificados raíz de confianza cuando el cliente de certificados sea administrado por más de un servidor de HPOM.

Certificados

HPE Operations Agent usa los dos tipos de certificados siguientes:

- Certificados raíz
- Certificados de nodo

Un certificado raíz es un certificado autofirmado que contiene la identidad de la autoridad de certificación del servidor de certificados. La clave privada correspondiente al certificado raíz es almacenada en el sistema del servidor de certificados y protegida contra acceso no autorizado. La autoridad de certificación usa su certificado raíz para firmar digitalmente todos los certificados.

Cada nodo del agente del entorno administrado recibe un certificado de nodo emitido por el servidor de certificados. Mientras se emite el certificado, el cliente de certificados que se ejecuta en el nodo del agente almacena la clave privada correspondiente en el sistema de archivos.

Nota: Un certificado de nodo contiene un identificador de nodo único—`OvCoreId`. A continuación se muestra un ejemplo de `OvCoreId: d498f286-aa97-4a31-b5c3-806e384fcf6e`

Cada nodo puede autenticarse de manera segura por medio de un certificado de nodo. El certificado de nodo puede ser verificado por otros nodos del entorno mediante el/los certificado-s raíz para verificar la firma. Los certificados nodo permiten establecer las conexiones basadas en SSL entre dos nodos HTTPS que usan autenticación de cliente y servidor, y pueden ser configurados para cifrar la comunicación.

La herramienta `ovcert` del cliente de certificados enuncia los contenidos del Key Store o muestra la información correspondiente a un certificado instalado.

Servidor de certificados de HPE Operations Agent

El servidor de certificados es responsable de las siguientes acciones:

- Crear e instalar certificados raíz autofirmados.
- Importar certificados raíz autofirmados del sistema de archivos.
- Almacenar claves privadas de certificados raíz.
- Conceder o denegar solicitudes de certificados.

- Crear un nuevo certificado y la clave privada correspondiente o crear una clave de instalación para la instalación de certificados manual.
- Ofrecer un servicio para el cliente que permita recuperar certificados raíz de confianza.

Autoridad de certificados

Nota: Los servidores de HPOM son configurados automáticamente como autoridad de certificados. El servidor de certificados predeterminado correspondiente a cada uno de los nodos del agente es el servidor de HPOM asociado al nodo.

La autoridad de certificados constituye una parte del servidor de certificados y es el centro de confianza de la administración de certificados. Los certificados firmados por la autoridad de certificados serán considerados certificados válidos y, por lo tanto, de confianza. La autoridad de certificación debe estar hospedada en una ubicación de alta seguridad. De manera predeterminada, la autoridad de certificación se encuentra instalada en el HPOM de hospedaje del sistema.

Puesto que la autoridad de certificación es la raíz de la confiabilidad, funciona mediante un certificado raíz autofirmado. Este certificado raíz y la clave privada correspondiente son creados y almacenados en el sistema de archivos con nivel de protección para permitir el funcionamiento de la autoridad de certificación. Tras la inicialización, la autoridad de certificados firma las solicitudes de certificados concedidas mediante su certificado raíz.

Cliente de certificados

El cliente de certificados se ejecuta en todos los sistemas del agente.

El cliente de certificados funciona del siguiente modo:

- El cliente de certificados comprueba si el nodo contiene un certificado válido.
- Si el nodo no contiene certificados, el cliente de certificados genera un nuevo par de claves pública y privada y crea una solicitud de certificado basada en una identidad única (valor `OvCoreId`) del nodo. El cliente de certificados envía la solicitud de certificado al servidor de certificados con propiedades de nodo adicionales y, a continuación, el cliente de certificados espera una respuesta.
- Las propiedades de nodo adicionales, como por ejemplo el nombre DNS y la dirección IP del nodo, ayudan al servidor de certificados a identificar el origen de la solicitud.

- Cuando el servidor de certificados emite un nuevo certificado, el cliente de certificados instala el certificado en el nodo. El cliente de certificados es responsable de asegurar que toda comunicación basada en HTTPS use este certificado.
- Si la solicitud no se procesa satisfactoriamente, se registrará un error descriptivo y se asignará el estado correspondiente.

El cliente de certificados realiza, además, las tareas siguientes:

- El cliente de certificados contacta con un servidor de certificados para actualizar los certificados raíz de confianza del servidor.
- Facilita la importación de certificados de nodo y las correspondientes claves privadas del sistema de archivos.
- Facilita la importación de certificados raíz de confianza.
- Proporciona información sobre el estado. El estado puede ser `OK`, `certificado válido`, `sin certificado`, `certificado solicitado` y `solicitud de certificado denegada`.

Implementación y actualización de certificados raíz

Puede ser necesario actualizar los certificados raíz de confianza de uno o más nodos, por ejemplo, en entornos que hospedan varios servidores de certificados.

Es posible transferir todos los certificados raíz de confianza a clientes de certificados de un modo seguro. Para ello suele bastar con transferir el certificado raíz de la autoridad de certificación. Sin embargo, puede ser necesario implementar uno o más certificados raíz adicionales para clientes de certificados determinados, por ejemplo, cuando existe más de una autoridad de certificación en un mismo entorno.

El cliente de certificados permite solicitar al servidor de certificados actualizar los certificados raíz de confianza por medio de la herramienta de línea de comando `ovcert`.

Parte III: Uso del componente HPE Operations Agent Performance Collection

HPE Operations Agent permite monitorizar un sistema recopilando métricas que indican el estado, rendimiento y disponibilidad de los elementos fundamentales del sistema.

El mecanismo de recopilación de datos del recopilador de datos de **oacore** está controlado por los ajustes del archivo **parm** ubicado en el directorio **%ovdatadir%** en Windows y en el directorio `/var/opt/perf` en UNIX o Linux. En función de las clases definidas en el archivo **parm**, el recopilador de datos de **oacore**, recopila un gran conjunto de datos así como información de la utilización de recursos en todo el sistema, como datos de proceso, datos de rendimiento para distintos dispositivos, datos de transacciones y datos de sistemas lógicos.

Los datos recopilados por **oacore** se almacenan en el Almacén de datos de métricas. El Almacén de datos de métricas es un almacén de datos basado en el sistema de gestión de bases de datos relacionales (RDBMS). Para cada clase de datos que se registra en el Almacén de datos de métricas, se crea un archivo de base de datos específico. Los archivos de base de datos están disponibles en `datadir/databases/oa`.

Puede usar herramientas como **Utility** y **Extract** para ver información específica almacenada en el Almacén de datos de métricas. 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`. El informe de examen enumera los parámetros de configuración del archivo **parm**, el nombre y la definición de cada aplicación, la adición, eliminación o cambio en las aplicaciones e información acerca de la disponibilidad del espacio en disco.

Puede usar el proceso de creación de líneas base para calcular y proporcionar valores de referencia en función de los datos históricos almacenados en el Almacén de datos de métricas. Loss datos de línea base ofrecen valores de referencia para analizar tendencias de rendimiento y establecer de manera dinámica valores óptimos del umbral para analizar el patrón de la utilización de recursos.

Capítulo 5: Administración de la recopilación de datos

HPE Operations Agent, con su recopilador de datos integrado, **oacore**, recopila continuamente los datos de estado y de rendimiento del sistema y almacena los datos recopilados en el Almacén de datos de métricas. Puede ver y analizar los datos registrados con herramientas como extract y utility. Se puede integrar HPE Operations Agent también con herramientas de análisis de datos como HP Performance Manager o HP Reporter para analizar los datos con ayuda de gráficos e informes.

El archivo de parámetros de configuración el archivo **parm** permite configurar el mecanismo de registro de datos predeterminado del recopilador de datos **oacore**. Puede usar el archivo **parm** para controlar el intervalo de registro de datos y el tipo de datos registrado.

Uso del almacén de datos de métricas

Con HPE Operations Agent versión 12.01, el Almacén de datos de métricas reemplaza el almacén de datos basado en el archivo de registro. Varios almacenes de datos como CODA y archivos de registro `Scope`, y DSI se han considerado como almacenes de datos basados en el sistema de gestión de bases de datos relacionales (RDBMS).

Después de actualizar HPE Operations Agent 11.xx a 12.00, los datos de HPE Operations Agent 11.xx (almacenados en los archivos de bases de datos CODA, archivos de registro `Scope` y archivos de registro DSI) se conservan en el modo de solo lectura y se guardan en:

```
/var/opt/perf/datafiles/
```

Puede acceder a los datos antiguos a través de utilidades como `ovcodautil`, extract, utility o a través de herramientas de informes como HP Performance Manager y HP Reporter.

Nota:

- Los datos antiguos de HPE Operations Agent 11.xx no migran al Almacén de datos de métricas.
- En un sistema HP-UX AR, si actualiza el HPE Operations Agent 11.xx en

12.xx, los datos de HPE Operations Agent 11.xx se guardan en:

`/var/opt/perf/datafiles/`

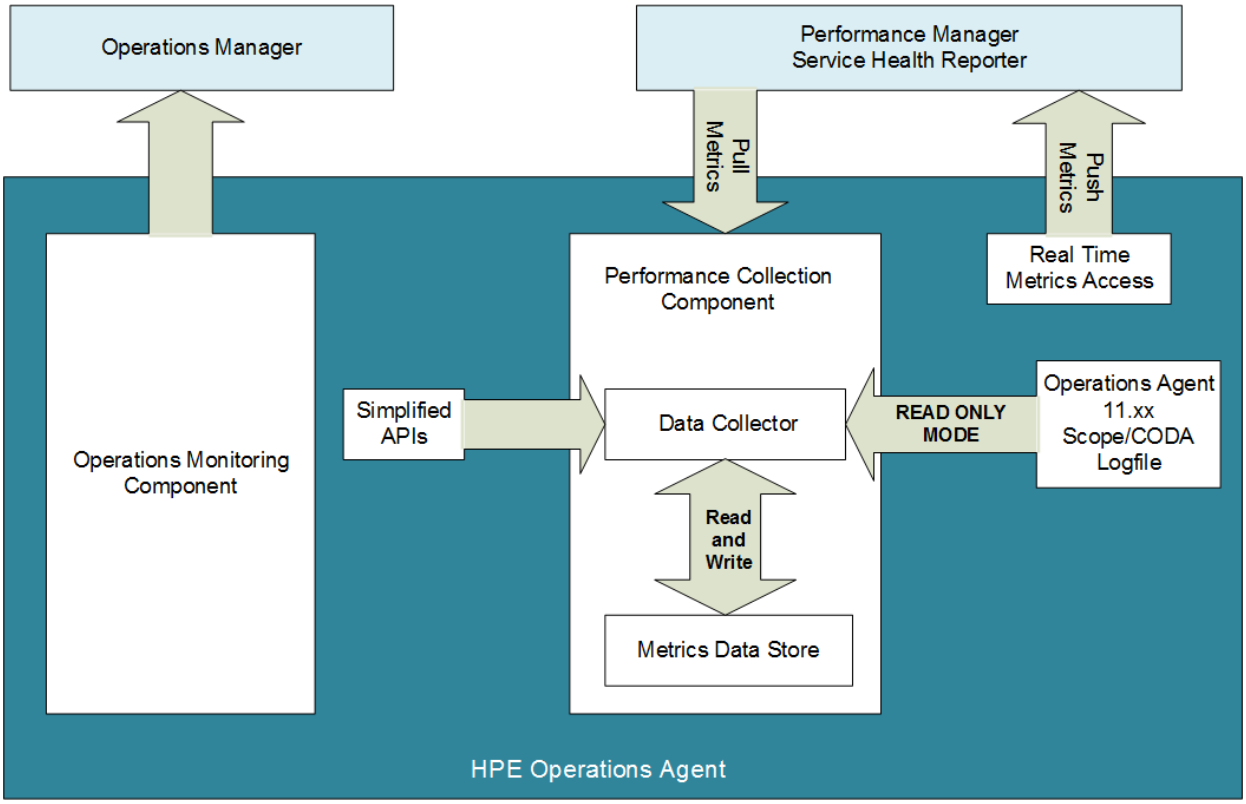
Puede acceder a estos datos con herramientas como `ovcodautil`, extract, utility o a través de herramientas de informes como HP Performance Manager y HP Reporter.

- En un sistema HP-UX IA, si actualiza el HPE Operations Agent 11.xx en 12.xx, los datos de HPE Operations Agent 11.xx se guardan en:

  `/var/opt/OV/tmp/BackUp/`

  No puede acceder a estos datos con herramientas como `ovcodautil`, extract, utility o a través de herramientas de informes como HP Performance Manager y HP Reporter.
- Después de actualizar HPE Operations Agent 11.xx a 12.xx, no se puede acceder a los datos del sistema lógico desde el HP Operations Agent 11.xx (guardado en los archivos de registro) no se puede acceder con herramientas como HP Performance Manager o HP Reporter.

La figura siguiente proporciona información general de la nueva arquitectura de HPE Operations Agent:



A pesar del cambio en el mecanismo de almacenamiento y recopilación de datos, el proceso de comparación del umbral a través de directivas no varía.

Recopilación de datos

Con la versión actual del HPE Operations Agent, los procesos **CODA** y **Scope** (**scopeux** en los nodos UNIX y Linux y **scopent** en los nodos Windows) se consolidan en un único proceso denominado **oacore.**

El recopilador de datos **oacore** recopila una gran serie de métricas de rendimiento del sistema que ofrecen una amplia visión del rendimiento y del estado del sistema. El recopilador de datos captura la información siguiente:

- Información relativa a la utilización de los recursos de todo el sistema.
- Datos de proceso.
- Datos de rendimiento de los dispositivos.
- Datos de transacción.
- Datos de sistemas lógicos.

El recopilador de datos de **oacore** recopila los datos en función de las clases de datos especificadas en el archivo **parm**. Las clases de métricas de rendimiento predeterminadas que puede definir en el archivo **parm** son global, application, process, disk device, lvolume, transation, configuration, netif, CPU, filesystem y host bus adapter.

Los datos recopilados por el recopilador de datos de **oacore** se almacenan en los archivos de base de datos específicos de la clase. Para cada clase de datos que se registra en el almacén de datos de métricas, se crea un archivo de base de datos. Los archivos de base de datos están disponibles en **datadir/databases/oa**.

Verificación del estado del proceso oacore

Si HPE Operations Agent se despliega desde HPOM, el proceso **oacore** se inicia automáticamente. Si HPE Operations Agent se instala en un nodo, el proceso **oacore** solo se inicia si están disponibles las licencias apropiadas en el nodo. Para obtener más información sobre las licencias, consulte la guía *HPE Operations Agent License Guide*.

El estado del proceso **oacore** se registra en el archivo `System.txt` situado en:

En Windows:

```
%ovdatadir%\log\System.txt
```

En Linux:

```
/var/opt/OV/log
```

Se agrega nueva información a este archivo cada vez que el recopilador de datos **oacore** se inicia, se detiene o cuando se encuentra una advertencia o error.

Una vez instalado HPE Operations Agent, siga los pasos para comprobar el estado del proceso **oacore**:

1. Inicie sesión en el nodo de HPE Operations Agent.
2. Ejecute el comando:

   En Windows x64:

   ```
   "%ovinstalldir%bin\win64\ovc"
   ```

   En Windows x86:

   ```
   "%ovinstalldir%bin\win32\ovc"
   ```

   En AIX:

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

   En HP-UX/Linux/Solaris:

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

Si el proceso **oacore** se ejecuta, obtendrá la salida siguiente:

```
oacore      Operations Agent Core              AGENT,OA     (25357)
Running
```

Inicio del proceso oacore

Si el proceso **oacore** no se está ejecutando, siga los pasos para iniciar el proceso **oacore**:

1. Inicie sesión en el nodo.
2. Vaya al directorio siguiente:

   En Windows x64:

   ```
   %ovinstalldir%bin\win64\
   ```

   En Windows x86:

   ```
   %ovinstalldir%bin
   ```

   En AIX:

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

   En HP-UX/Linux/Solaris:

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

3. Ejecute el comando siguiente para iniciar el proceso:

   ```
   ovc -start oacore
   ```

Nota: Para detener el proceso **oacore**, ejecute el comando siguiente:

```
ovc -stop oacore
```

Nota: **oacore** completa las solicitudes de acceso a los datos mucho antes que tenga lugar el agotamiento del tiempo de espera. Las solicitudes de acceso a los datos se completan en el orden en que llegan.

Si solicita muchos datos (por ejemplo, dos millones de registros de procesos), **oacore** puede tardar más tiempo de lo usual en completar la solicitud. Mientras **oacore** está procesando una solicitud de acceso a una gran cantidad de datos, otras solicitudes (como dsilog, extract) pueden agotar el tiempo de espera. Si obtiene un error de agotamiento de tiempo de espera, debe volver a realizar la solicitud.

Verificación del registro de datos

Los datos de métricas recopilados por el recopilador de datos **oacore** se guardan en el Almacén de datos de métricas. Sigue los pasos para verificar el registro de datos:

1. Inicie sesión en el nodo.
2. Ejecute el comando:

   En Windows x64:

   ```
   %ovinstalldir%bin\win64\ovcodautil –obj
   ```

   En Windows x86:

   ```
   %ovinstalldir%bin\ovcodautil –obj
   ```

   En AIX:

   ```
   /usr/lpp/OV/bin/ovcodautil –obj
   ```

   En HP-UX/Linux/Solaris:

   ```
   /opt/OV/bin/ovcodautil –obj
   ```

Si se están registrando los datos, se muestra una lista de todas las métricas registradas.

Control del espacio en disco usado por los archivos de base de datos

Performance Collection Component incluye la administración automática de los archivos de base de datos.

Control del espacio en disco usado por los archivos de bases de datos que almacenan las clases de métricas de rendimiento predeterminadas

El tamaño de los archivos de bases de datos que almacenan la *clase de métricas de rendimiento predeterminadas*, depende del tamaño máximo especificado en el archivo **parm**. Si se cambia la especificación de tamaño del archivo **parm**, **oacore** lo detecta *solo* durante el inicio.

Los archivos de base de datos de una clase se sustituyen cuando se alcanza el tamaño máximo especificado en el archivo **parm**. Durante una sustitución, el 20 por ciento de los datos más antiguos se suprimen. Si el *tamaño* no se especifica en el archivo **parm**, los archivos de base de datos se sustituyen cuando se alcanza el tamaño máximo de 1 GB.

Nota: El tamaño de los archivos de base de datos no se puede establecer por debajo de 1 MB.

Cambio del tamaño máximo especificado en el archivo parm

Para cada clase de datos que se registra en el almacén de datos, se crean cinco archivos de base de datos. El tamaño de estos archivos de bases de datos depende del tamaño máximo especificado en el archivo **parm**.

Nota: Para cada clase de datos que se registra en el almacén de datos, se crea un archivo de base de datos inicialmente. Cuando el primer archivo de base de datos alcanza un tamaño aproximadamente igual al 20% del tamaño máximo especificado en el archivo **parm**, se crea el segundo archivo de base de datos. Este proceso continúa hasta que se creen cinco archivos de bases de datos.

Si se cambia la especificación de tamaño del archivo **parm**, **oacore** lo detecta durante el inicio. El nuevo tamaño solo es efectivo para los nuevos archivos de bases de datos. Eventualmente cuando los archivos de base de datos más antiguos se suprimen durante la sustitución, se crean los nuevos archivos de base

de datos que tengan un tamaño aproximadamente igual al 20% del tamaño máximo especificado en el archivo **parm**.

Nota: El tamaño total puede diferir, o puede ser mayor o menor, del tamaño configurado, ya que el nuevo tamaño solo es efectivo para los nuevos archivos de bases de datos. Gradualmente, cuando los archivos de bases de datos más antiguos se suprimen durante la sustitución, el tamaño total puede coincidir el tamaño especificado en el archivo **parm**.

Por ejemplo:

Si especifica 10 MB como el tamaño máximo de una clase de datos en el archivo **parm**, se crean cinco archivos de 2 MB.

Durante la sustitución, cuando se alcanza el tamaño máximo (10 MB) especificado en el archivo **parm**, se suprime el archivo de base de datos más antiguo (2 MB de tamaño).

Escenario 1: El tamaño máximo disminuye de 10 MB a 5 MB:

Debe reiniciar el componente Performance Collection para que los cambios surtan efecto.

Todos los archivos de base de datos existentes siguen teniendo el tamaño de 2 MB.

Finalmente como los archivos de la base de datos más antiguos se suprimen durante la sustitución, se crean nuevos archivos de base de datos con un tamaño máximo de 1 MB.

Después de que se hayan suprimido todos los archivos antiguos (los que existían antes del reinicio), se crearán cinco archivos de base de datos cada uno de ellos con un tamaño máximo de 1 MB. Por tanto, se conseguirá el tamaño máximo de 5 MB especificado en el archivo **parm**.

Nota: El tamaño total puede exceder del tamaño configurado, ya que el nuevo tamaño solo es efectivo para los nuevos archivos de bases de datos. Gradualmente, cuando los archivos de bases de datos más antiguos se suprimen (durante la sustitución), el tamaño total puede coincidir el tamaño especificado en el archivo **parm**.

Escenario 2: El tamaño máximo aumenta de 10 MB a 15 MB:

Debe reiniciar el componente Performance Collection para que los cambios surtan efecto.

Todos los archivos de base de datos existentes siguen teniendo el tamaño de 2 MB.

Finalmente como los archivos de la base de datos más antiguos se suprimen durante la sustitución, se crean nuevos archivos de base de datos con un tamaño máximo de 3 MB.

Después de que se hayan suprimido todos los archivos antiguos (los que existían antes del reinicio), se crearán cinco archivos de base de datos cada uno de ellos con un tamaño máximo de 3 MB. Por tanto, se conseguirá el tamaño máximo de 15 MB especificado en el archivo **parm**.

Nota: El tamaño total puede ser inferior al tamaño configurado, ya que el nuevo tamaño solo es efectivo para los nuevos archivos de bases de datos. Gradualmente, cuando los archivos más antiguos se suprimen (durante la sustitución), el tamaño total puede coincidir el tamaño especificado en el archivo **parm**.

Control del espacio en disco usado por los archivos de bases de datos que almacenan los datos personalizados

El tamaño máximo de los archivos de base de datos que almacenan los *datos personalizados* se establece en 1 GB de manera predeterminada. Este tamaño no se puede configurar.

Para cada clase de datos que se registra en el almacén de datos, se crea un archivo de base de datos inicialmente. Cuando el primer archivo de base de datos alcanza un tamaño aproximadamente igual al 20% de 1 GB (200 MB), se crea el segundo archivo de base de datos. Este proceso continúa hasta que se creen cinco archivos de bases de datos, cada uno de ellos de 200 MB. Los archivos de base de datos se sustituyen cuando se alcanza el tamaño máximo de 1 GB. Durante una sustitución, el 20 por ciento de los datos más antiguos se suprimirán (el archivo más antiguo de la base de datos se elimina de manera permanente).

Nota:

- En los sistemas UNIX, el proceso **oacore** requiere recursos adecuados como el descriptor de archivo abiertos. En la mayoría de los sistemas UNIX, este límite está configurado en 256 (ulimit -n). Para el proceso **oacore**, este límite es suficiente para funcionar en los escenarios habituales. En los sistemas configurados para registrar clases personalizadas adicionales, el descriptor de archivo abiertos debe configurarse proporcionalmente.

Por ejemplo:

Si tiene unas 20 clases gestionadas, se recomienda configurar el descriptor de archivos abiertos a 512.

Si tiene unas 50 clases gestionadas, se recomienda configurar el descriptor de archivos abiertos a 1024.

Después de configurar el descriptor de archivos abiertos, reinicie el proceso **oacore**.

- En las plataformas Solaris, el comando `ovc -start oacore` inicia el script `/opt/perf/bin/runoacore`. Este script establece el límite del descriptor de archivos abiertos en 4096 antes de iniciar el proceso **oacore**.

Detención y reinicio de la recopilación de datos

El proceso **oacore** 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, **ttd.conf**. (Para obtener más información, consulte ¿Qué es el seguimiento de transacciones?)
- Está modificando rangos de distribución u objetivos de nivel de servicio (SLO) en el archivo de configuración de transacciones, **ttd.conf**. (Para más información sobre el seguimiento de transacciones, consulte ¿Qué es el seguimiento de transacciones?)
- Está modificando el archivo **parm** y desea que los cambios surtan efecto. Los cambios realizados en el archivo **parm** surten efecto solo cuando se reinicia el proceso **oacore**.
- Está apagando el sistema.
- Está agregando hardware o modificando los cambios de configuración. Los cambios realizados surten efecto solo cuando se reinicia el proceso **oacore**.

Detención de la recopilación de datos

La opción stop del script **ovpa** asegura que no se pierde ningún dato cuando se detienen **oacore** y otros procesos de Componente Performance Collection.

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

En Windows:

```
%ovinstalldir%bin\ovpacmd stop
```

En HP-UX/Linux/Solaris:

```
/opt/perf/bin/ovpa -stop
```

En AIX:

```
/usr/lpp/perf/bin/ovpa -stop
```

Nota: oacore 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 **oacore** y los restantes procesos de Componente Performance Collection una vez apagado el sistema, o después de haberlos detenido, se utilice `<InstallDir>/ovpa start`. En este caso, InstallDires el directorio en donde se ha instalado Componente Performance Collection.

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

Cuando reinicia **oacore**, el componente Performance Collection continúa usando los mismos archivos de base de datos usados antes de detener el proceso **oacore**. Se anexan nuevos registros al final de los archivos existentes.

Nota: 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, si la hora del sistema adelanta una hora, el registro de datos continúa sin cambios. Si la hora del sistema retrocede una hora, no se registran datos durante una hora hasta que la hora del sistema se sincronice con la marca de fecha y hora de los últimos datos registrados.

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

Si adelanta una hora en la hora del sistema, el registro de datos continúa sin cambios. Si retrocede una hora en la hora del sistema, **oacore** deja de registrar durante una hora hasta que la hora del sistema se sincronice con la marca de fecha y hora de los últimos datos registrados.

Nota: La hora del sistema debe sincronizarse con la hora de la recopilación de datos para evitar desfases en la recopilación.

Uso del archivo parm

El archivo **parm** es un archivo de texto que contiene las instrucciones para que el recopilador de datos **oacore** registre datos específicos de rendimiento.

Cuando instala HPE Operations Agent 12.00 en un nodo donde la versión anterior de HPE Operations Agent no está instalada, el archivo **parm** se sitúa en dos directorios diferentes. Durante una actualización de versiones anteriores de HPE Operations Agent a la versión actual, se actualiza el archivo **parm** situado en uno de los directorios. Para obtener más información, consulte la disponibilidad del archivo **parm** en los siguientes escenarios:

- Instalación de HPE Operations Agent 12.xx
- Actualización a HPE Operations Agent 12.xx

Instalación de HPE Operations Agent 12.xx

Cuando instala HPE Operations Agent 12.xx en un nodo donde la versión anterior de HPE Operations Agent no está instalada, el archivo **parm** se sitúa en dos directorios diferentes.

En Windows:

```
%ovinstalldir%\newconfig
```

```
%ovdatadir%
```

Nota: En Windows, el archivo **parm** existe con la extensión .mwc (**parm.mwc**).

En HP-UX/Linux/Solaris:

```
/opt/perf/newconfig
```

```
/var/opt/perf
```

En AIX:

```
/usr/lpp/perf/newconfig
```

```
/var/opt/perf
```

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

Para 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).

Actualización a HPE Operations Agent 12.xx

En Windows

Cuando actualiza HPE Operations Agent en un nodo, el proceso de actualización actualiza la copia del archivo **parm** disponible en el directorio **newconfig**. El archivo **parm** que reside en el directorio *%ovdatadir%* permanece sin modificar y continúa rigiendo el mecanismo de recopilación de datos en el nodo. Este método permite mantener el mecanismo de recopilación de datos configurados después de una actualización.

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. Para obtener más información sobre cómo modificar el archivo **parm**, consulte "Modificación del archivo parm" en la página siguiente

En UNIX/Linux

En todos los sistemas UNIX/Linux, se utiliza la utilidad del procesador de macros m4 para preprocesar el archivo **parm**. El Componente Performance Collection preprocesa el archivo **parm** que se encuentra en el directorio **/var/opt/perf** para crear un archivo **parm** en tiempo de ejecución que es un archivo de texto.

Puede usar el comando siguiente para generar una archivo **parm** en tiempo de ejecución:

 # m4 -DPARMOS=<sistema_operativo>/var/opt/perf/parm > parm.m4

Establezca el valor apropiado para *<sistema_operativo>*.

Nota: El valor de <sistema_operativo> distingue entre mayúsculas y minúsculas. Para obtener un valor apropiado para el sistema operativo, ejecute el comando siguiente:

`uname`

Por ejemplo:

Si ejecuta el comando `uname` en un sistema operativo Linux, se genera la salida siguiente:

Linux

Los archivos **parm** preexistentes, sin la utilidad m4, funcionan para las versiones anteriores de HPE Operations Agent.

El sistema busca m4 durante la instalación. Si m4 no está presente, asegúrese de que instala m4 en el sistema.

Para obtener más información, consulte la sección *Prerequisites for Installing HPE Operations Agent* en la guía *HPE Operations Agent and HPE Operations Smart Plug-in for Infrastructure Installation Guide*.

Nota: Puede agregar definiciones de aplicación personalizadas al archivo externo e incluirlo en el archivo **parm** usando el comando *#include (var/opt/perf/parm.apps)*. Para obtener más información sobre cómo definir una aplicación, consulte "Parámetros de definiciones de aplicación" en la página 187

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 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/Solaris:

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

Nota: Puede usar el comando `/opt/perf/bin/ovpa -restart scope` para reiniciar Componente Performance Collection. Este comando solo se conserva por retrocompatibilidad después de la actualización de las versiones anteriores a HPE Operations Agent 12.xx.

En AIX:

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

Nota: Puede usar el comando `/usr/lpp/perf/bin/ovpa -restart scope` para reiniciar Componente Performance Collection.Este comando solo se conserva por retrocompatibilidad después de haberse actualizado de las versiones anteriores a HPE Operations Agent 12.xx.

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

En Windows:

```
%ovinstalldir%bin\ovpacmd stop RTMA
```

```
%ovinstalldir%bin\ovpacmd start RTMA
```

En HP-UX/Linux/Solaris:

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

En AIX:

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

Parámetros del archivo parm

El recopilador de datos - **oacore** está controlado por parámetros específicos en el archivo de parámetros de la recopilación (**parm**) que hacen lo siguiente:

- 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 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.

Se pueden modificar estos parámetros para configurar **oacore** con objeto de registrar los datos del rendimiento que coincidan con los requisitos del sistema monitorizado (consulte Modificación del archivo parm).

Nota: La recopilación de parámetros de configuración del sistema solo tiene lugar cuando se inician, agregan o suprimen dispositivos del sistema.

oacore utiliza los parámetros del archivo **parm** enumerados en la tabla siguiente. 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 y Parámetros de definiciones de aplicación.

Parámetros del archivo parm utilizados por oacore

Parámetro	Valores u opciones
Descripciones de los parámetros	
id	Identificador del sistema
log	global application [=all] - permite a **oacore** registrar todas las aplicaciones en el archivo logappl en cada intervalo, con independencia de si las aplicaciones están activas o no. application [=prm] (sólo en HPE-UX) - permite a **oacore** registrar grupos de Process Resource Manager (PRM) activos en el almacén de datos. process device=disk,lvm,cpu,filesystem,all (lvm sólo en HP-UX y Linux) transaction=correlator, resource (resource sólo en HP-UX) **Nota:** Solo las métricas de clase de nivel superior se registran después de actualizar de versiones anteriores a HPE Operations Agent 12.00. logicalsystem (En el caso de Solaris, se admite el sistema lógico en el entorno operativo Solaris 10 o posterior) En AIX, el sistema lógico se admite en sólo en el entorno global LPAR en AIX 5L V5.3 ML3 y posterior y en WPAR en AIX 6.1 TL2. Para habilitar el registro lpar, logicalsystems=lpar logicalsystems Para habilitar el registro wpar, logicalsystems=wpar

Parámetros del archivo parm utilizados por oacore, continuación

Parámetro	Valores u opciones
	Para habilitar tanto el registro lpar como el wpar, logicalsystems=lpar,wpar logicalsystems=wpar,lpar logicalsystems=all
subprocinterval	valor en segundos (no en HP-UX)
javaarg	True|False
procthreshold (igual que threshold)	cpu=*porcentaje* disk=*velocidad* (no en Linux ni Windows) memory=nn (valores en MB) IO=*velocidad* (valores en KBs/sec) nonew nokilled shortlived
appthreshold	cpu=*porcentaje*
diskthreshold	util=*velocidad*
bynetifthreshold	iorate=*velocidad*
fsthreshold	util=*velocidad*
lvthreshold	iorate=*velocidad*
bycputhreshold	cpu=*porcentaje*
fstypes	Para incluir sólo sistemas de archivos específicos para el registro de datos, use la sintaxis *<sistema_de_archivos1>, <sistema_de_archivos2>, <sistema_de_archivos3>, ...* Para excluir un sistema de archivos, use la sintaxis !<*sistema de archivos*>.
wait	cpu=*porcentaje* (sólo HP-UX) disk=*porcentaje* (sólo HP-UX)

Parámetros del archivo parm utilizados por oacore, continuación

Parámetro	Valores u opciones
	mem=*porcentaje* (sólo HP-UX) sem=*porcentaje* (sólo HP-UX) lan=*porcentaje* (sólo HP-UX)
`size`	tamaño (valores en MB) process=nn (el valor máximo es 4096) El valor máximo para todas las clases siguientes es 2048. global=nn application=nn device=nn transaction=nn logicalsystem=nn
`collectioninterval`	`process=ss (valores en segundos)` `global=ss`

`gapapp`	`blank`
	`unassignedprocesses`
	`existingapplicationname`
	`other`

`Flush`	ss (valores en segundos) 0 (deshabilita el vaciado de datos)

`project_app`
NOTE: Sólo en Solaris
	`true`
	`false`
	`(sólo en Solaris 10 y superior)`

`proccmd`	0 (deshabilita el registro de los comandos de proceso) 1 (habilita el registro de los comandos de proceso) La longitud máxima del comando del proceso registrado es siempre 4096.
`proclist` **NOTE:** Sólo en Solaris	all (el Componente Performance Collection de la zona global monitoriza todos los procesos que se ejecutan en las zonas global y no global) local (el Componente Performance Collection de la zona

Parámetros del archivo parm utilizados por oacore, continuación

Parámetro	Valores u opciones
	global monitoriza sólo los procesos que se ejecutan en la zona global) Este parámetro no tiene efecto en las zonas no globales.
appproc **NOTE:** Sólo en Solaris	all (configura Componente Performance Collection para que calcule métricas APP_ con procesos para aplicaciones que pertenecen a las zonas global y no global) local (configura Componente Performance Collection para que calcule métricas APP_ con procesos para aplicaciones que pertenecen sólo a la zona global) Este parámetro no tiene efecto en las zonas no globales.
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)` **NOTE:** Este parámetro no afecta a HP-UX. Para conmutar entre los modos anteriores, es preciso ejecutar el comando **midaemon -ignore_mt** en HP-UX. Para más información, consulte Registro de métricas calculadas con la normalización basada en núcleo.
cachemem	f o free (HPE Operations Agent no incluye el tamaño de la caché del búfer al calcular el valor de la métrica GBL_MEM_UTIL) u o usuario (HPE Operations Agent incluye el tamaño de la caché del búfer al calcular el valor de la métrica GBL_MEM_UTIL)
Parámetros de definiciones de aplicación	
application	*nombre de la aplicación*

Parámetros del archivo parm utilizados por oacore, continuación

Parámetro	Valores u opciones
`file`	*nombre de archivo [, ...]*
`argv1`	primer argumento del comando [,]
`cmd`	expresión regular de la línea de comandos
`user`	*nombre de inicio de sesión del usuario [,]*
`group`	*nombre de grupo [,]*
`or`	
`priority`	valor bajo-valor alto (el rango varía en función de la plataforma)

Descripciones de los parámetros

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

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 el almacén de datos 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 **oacore** va a recopilar.

- **log global** permite a **oacore** 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 **oacore** inscribir registros de aplicación activos en el almacén de datos. De manera predeterminada, **oacore** registra solo las aplicaciones que tienen procesos activos durante un intervalo.

Este parámetro `log application=all` en el archivo **parm** permite a **oacore** 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).

Solo en HP-UX, puede especificar el parámetro `log application=prm` para permitir que **oacore** registre los grupos de Process Resource Manager (PRM) en el archivo logappl. Si se especifica este parámetro, **oacore** 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 **oacore** registrar información sobre procesos interesantes en el almacén de datos. 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 **oacore** registrar la información sobre discos individuales, volúmenes lógicos (solo HP-UX), CPU y sistemas de archivo en el almacén de datos.

  Nota: Use `lvm` solo en sistemas que se ejecutan con el sistema operativo HP-UX o Linux.

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

  Nota: De manera predeterminada, los registros netif se almacenan aunque no se especifique en el archivo **parm**.

  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, netif
  ```

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 **oacore** 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.

- **log transaction** permite a **oacore** registrar transacciones de ARM en el almacén de datos. Para habilitar **oacore** para recopilar datos, debe ejecutarse en el sistema un proceso que esté instrumentado con la API de Application Response Measurement (ARM).

  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**.

  Nota: Cuando actualiza a HPE Operations Agent 12.xx, la transacción del registro permite a **oacore** registrar solo métricas de clase de transacciones de nivel superior.

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

  Nota: Las métricas BYLS no se admiten en las plataformas Windows y Linux.

  HPE Operations Agent 12.xx no recopila las métricas BYLS de Xen, KVM, VMware vSphere, Hyper-V y otros dominios de virtualización.

  En AIX 6.1 TL2, el registro de BYLS para LPAR y WPAR se puede configurar usando el parámetro logicalsystems del archivo **parm**. Para obtener más información, consulte "Parámetros del archivo parm utilizados por oacore" en la página 166.

Nota: Si se especifica el parámetro log sin opciones, **oacore** registra sólo los datos globales y de proceso.

Umbrales

Los parámetros de umbral permiten a **oacore** registrar solo la información crítica en el almacén de datos y descartar los detalles no necesarios y no críticos del

sistema.

Si el valor de umbral especificado excede para una instancia particular de una clase de datos, **oacore** inscribe un registro para esa instancia. Se pueden especificar valores inferiores para el umbral con objeto de permitir que **oacore** registre más datos o se pueden especificar valores superiores para el umbral con el fin de que **oacore** registre menos datos.

Los parámetros siguientes especifican los umbrales para distintas clases de métricas:

- Procthreshold
- appthreshold
- 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` solo afecta a los procesos que están registrados y no afecta 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

`cpu`	Establece el porcentaje de utilización de CPU que debe exceder un proceso para convertirse en "interesante" y registrarse. El valor percent es un número real que indica el uso de CPU general. **Por ejemplo:** cpu=7.5 indica que un proceso se registra si excede del 7,5 por ciento de utilización de la CPU en una muestra de 1 minuto.
`disk`	Establece la velocidad de E/S del disco físico por segundo que debe exceder un proceso para convertirse en "interesante" y

Opciones de procthreshold para datos de proceso , continuación

	registrarse. El valor es un número real. **Nota:** Esta opción no está disponible en Linux o Windows. **Por ejemplo:** disk=8.0 indica que un proceso se registrará si la velocidad promedio de E/S del disco físico excede de 8 KB por segundos.
`memory`	Establece el umbral de memoria que debe exceder un proceso para convertirse en "interesante" y registrarse. 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 PROC_MEM_VIRT. 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.
`IO`	Establece el umbral de E/S que debe exceder un proceso para convertirse en "interesante" y registrarse. El valor se da en unidades de kilobyte. **Por ejemplo:** IO=100 indica que un proceso se registrará si la tasa de E/S del proceso (PROC_IO_BYTE_RATE) excede los 100 KBs por segundo.
`nonew`	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 `shortlived` *no* se especifica, se registran solo los nuevos procesos que duran más de un segundo.
`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 `shortlived` *no* se especifica, se registran solo

Opciones de procthreshold para datos de proceso , continuación

`shortlived`	Permite el registro de procesos que se ejecutan durante menos de un segundo en un intervalo. Esto suele aumentar de manera importante el número de procesos registrados. Si **oacore** encuentra el umbral efímero en el archivo **parm**, registra los procesos **shortlived**, con independencia del umbral de la CPU o de disco, siempre que las opciones `nonew` y `nokilled` se supriman. De manera predeterminada, no se registra ningún proceso `shortlived`.

`procthreshold` especifica los umbrales para la clase PROCESS. Los valores predeterminados son los siguientes:

- 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

appthreshold

El parámetro `appthreshold` 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 esta se registre en el almacén de datos.

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

diskthreshold

El parámetro `appthreshold` 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 **oacore** 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 **oacore** 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, **oacore** 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 **oacore** 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 **oacore** 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 umbral 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 **oacore** registrar los detalles de las clases CPU que están ocupadas más del 90% del tiempo.

subprocinterval

El parámetro `subprocinterval`, si se especifica, invalida el valor predeterminado que utiliza **oacore** 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, **oacore** 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 195.

En algunos sistemas con miles de subprocesos o procesos activos, `subprocinterval` debería aumentarse para reducir la sobrecarga general de **oacore**. 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 **oacore**. Esta configuración puede adoptar valores que son factores del intervalo de registro del proceso, tal como se especificó en el archivo **parm**.

Nota: 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 agrega 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.

fstypes

El parámetro `fstypes` permite monitorizar sistemas de archivo concretos en el sistema. De manera predeterminada, el recopilador **oacore** registra datos de todos los sistemas de archivos vinculados al sistema. Con el parámetro `fstypes` se puede habilitar la recopilación de datos sólo para sistemas de archivos concretos.

En el archivo **parm**, es preciso seleccionar en el parámetro fstypes el nombre del sistema de archivos que se desea monitorizar. Se pueden especificar varios nombres de sistemas se archivos separados por comas.

Ésta es la sintaxis de este parámetro:

`fstypes=<sistema_de_archivos1>,<sistema_de_archivos2>, ...`

`<file_system1>` y `<file_system2>` son los tipos de sistema de archivos.

En HP-UX, Linux y AIX, todos los sistemas de archivos disponibles con los tipos de sistemas de archivos se muestran en el archivo **/etc/fstab**.

En AIX, los sistemas de archivos disponibles con tipos de sistemas de archivos se muestran en el archivo **/etc/filesystems**.

En Solaris, los sistemas de archivos disponibles con tipos de sistemas de archivos se muestran en el archivo **/etc/vfstab**.

Para averiguar el tipo de sistema de archivos en Windows, haga clic con el botón secundario en la unidad de disco en el Explorador de Windows y haga clic en **Propiedades**. El valor que se muestra en Sistema de archivos es el tipo de sistema de archivos que se puede especificar con el parámetro `fstypes`.

En este parámetro se puede seleccionar una lista de inclusión o una lista de exclusión. Una lista separadas por comas de los tipos de sistemas de archivos indica una lista de inclusión y permite a Agent monitorizar los datos sólo en los sistemas de archivos especificados.

Al configurar el parámetro `fstypes` , use los nombres de los sistemas de archivos que devuelven los comandos del sistema operativo. En este parámetro también se puede seleccionar una lista de inclusión, que es una lista de tipos de sistemas de archivos separada por comas que comienza con el carácter `!`. Al especificar el carácter `!` al principio de la lista se garantiza que Agent no monitoriza los sistemas de archivos que pertenecen a la lista.

Ejemplo 1:

```
fstypes = tmpfs, mvfs, nfs
```

Ejemplo 2:

```
fstypes = !tmpfs, mvfs
```

Ejemplo 3:

```
fstypes =
```

o

```
fstypes = *
```

Al especificar el carácter `*` o `blank`, se asegura de que HPE Operations Agent monitoriza todos los sistemas de archivos que están disponibles.

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 almacén de datos. Consulte "Parámetros del archivo parm utilizados por oacore" en la página 166 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 los archivos de base de datos. El recopilador **oacore** lee estas especificaciones cuando se inicia.

Los archivos de base de datos que se almacenan en las *clases de métrica de rendimiento predeterminadas* se sustituyen cuando se alcanza el tamaño máximo especificado en el archivo **parm**. Si el tamaño no se especifica en el archivo **parm**, los archivos de base de datos se sustituyen cuando se alcanza el tamaño máximo de 1 GB.

El tamaño máximo de los archivos de base de datos que almacenan los *datos personalizados* se establece en 1 GB de manera predeterminada. Los archivos de base de datos se sustituyen cuando se alcanza el tamaño máximo de 1 GB.

Para obtener más información sobre cómo controlar el espacio en disco usado por archivos de bases de datos, consulte Control del espacio en disco usado por los archivos de bases de datos.

javaarg

El parámetro `javaarg` es un indicador que solo afecta al valor de la métrica **proc_proc_argv1**. De manera predeterminada, este parámetro se establece en true.

Cuando `javaarg` es true y la métrica proc_proc_argv1 se registra con el nombre clase o jar. Este sistema es muy útil para definir aplicaciones específicas para Java. Cuando hay un nombre de clase en la cadena de comandos para un proceso, puede usar el calificador argv1=application para definir la aplicación por nombre de clase.

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

Por ejemplo:

Cuando se ejecuta el proceso siguiente:

```
java -XX:MaxPermSize=128m -XX:+CMSClassUnloadingEnabled -Xms8000m
-Xmx8000m -Dserver_port=1099 -jar ./ApacheJMeter.jar
```

- Si `javaarg` se establece en **True**, el valor de proc_proc_argv1 será -jar ./ApacheJMeter.jar
- Si `javaarg` se establece en **False**, el valor de proc_proc_argv1 será -jar -XX:MaxPermSize=128m

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, **oacore** no registrará ningún dato de

aplicación o dispositivo que no alcance los umbrales especificados en el archivo **parm**.

project_app

Si establece el parámetro `project_app` en True, Componente Performance Collection considera que cada uno de los proyectos de Solaris es una aplicación (y que el Id. del proyecto es el Id. de la aplicación). Para ignorar proyectos de Solaris, seleccione False en este parámetro.

Nota: El parámetro `project_app` solo es compatible con Solaris 10 y las versiones superiores.

proclist

Este parámetro sólo se puede usar en zonas globales de Solaris; no tiene efecto en el HPE Operations Agent que se ejecuta en una zona no global.

En una zona global, si en este parámetro se selecciona `all`, Componente Performance Collection monitoriza todos los procesos de las zonas globales y no globales. Para monitorizar sólo los procesos que pertenecen a la zona global, seleccione `local` en este parámetro.

appproc

El parámetro `appproc` solo se encuentra disponible en Solaris. Este parámetro sólo se puede usar en una zona global; no tiene efecto en el HPE Operations Agent que se ejecuta en una zona no global.

En una zona global, si en este parámetro se elige el valor **`all`**, Componente Performance Collection incluye los procesos de las aplicaciones de las zonas global y no global al calcular los valores de todas las métricas APP_. Para incluir sólo las aplicaciones de la zona global para el cálculo de las métricas APP_, seleccione **local** en este parámetro.

proccmd

El parámetro `proccmd` permite el registro de los comandos de proceso en el almacén de datos de . De manera predeterminada, el valor de este proceso se establece en 0 y se deshabilita el registro de los comandos del proceso. Para habilitar el registro de los comandos del proceso, establezca el valor de este parámetro en 1.

Nota: El registro del parámetro `proccmd` se activa cuando el valor de este parámetro es mayor o igual que 1. La longitud del comando del proceso

registrado es siempre 4096, con independencia del valor especificado en este parámetro.

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.

En las máquinas Linux, este parámetro se establece en false de manera predeterminada.

Este parámetro no afecta a HP-UX. Para conmutar entre los modos, es preciso ejecutar el comando `midaemon -ignore_mt` en HP-UX. Para obtener más información, consulte "Registro de métricas calculadas con la normalización basada en núcleo" en la página 203.

Cuando el indicador `ignore_mt` se establece como un parámetro de línea de comandos en `midaemon` en los sistemas HP-UX, algunas métricas se ven afectadas. Para obtener más información sobre la lista de métricas que se ven afectadas, consulte "Archivo parm" en la página 410.

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**.

Nota: 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.

Métricas afectadas al configurar el indicador ignore_mt en midaemon en los sistemas HP-UX

Clase de métrica global	
GBL_CPU_TOTAL_UTIL	GBL_CPU_CSWITCH_UTIL_CUM
GBL_CPU_TOTAL_UTIL_CUM	GBL_CPU_CSWITCH_UTIL_HIGH

GBL_CPU_TOTAL_UTIL_HIGH	GBL_CPU_CSWITCH_TIME
GBL_CPU_TOTAL_TIME	GBL_CPU_CSWITCH_TIME_CUM
GBL_CPU_TOTAL_TIME_CUM	GBL_CPU_INTERRUPT_UTIL
GBL_CPU_SYS_MODE_UTIL	GBL_CPU_INTERRUPT_UTIL_CUM
GBL_CPU_SYS_MODE_UTIL_CUM	GBL_CPU_INTERRUPT_UTIL_HIGH
GBL_CPU_SYS_MODE_TIME	GBL_CPU_INTERRUPT_TIME
GBL_CPU_SYS_MODE_TIME_CUM	GBL_CPU_INTERRUPT_TIME_CUM
GBL_CPU_TRAP_TIME	GBL_CPU_VFAULT_TIME
GBL_CPU_TRAP_TIME_CUM	GBL_CPU_VFAULT_TIME_CUM
GBL_CPU_TRAP_UTIL	GBL_CPU_VFAULT_UTIL
GBL_CPU_TRAP_UTIL_CUM	GBL_CPU_VFAULT_UTIL_CUM
GBL_CPU_TRAP_UTIL_HIGH	GBL_CPU_VFAULT_UTIL_HIGH
GBL_CPU_USER_MODE_TIME	GBL_CPU_IDLE_UTIL
GBL_CPU_USER_MODE_TIME_CUM	GBL_CPU_IDLE_UTIL_CUM
GBL_CPU_USER_MODE_UTIL	GBL_CPU_IDLE_UTIL_HIGH
GBL_CPU_USER_MODE_UTIL_CUM	GBL_CPU_IDLE_TIME
GBL_CPU_NICE_UTIL	GBL_CPU_IDLE_TIME_CUM
GBL_CPU_NICE_UTIL_CUM	GBL_CPU_NORMAL_UTIL
GBL_CPU_NICE_UTIL_HIGH	GBL_CPU_NORMAL_UTIL_CUM
GBL_CPU_NICE_TIME	GBL_CPU_NORMAL_UTIL_HIGH
GBL_CPU_NICE_TIME_CUM	GBL_CPU_NORMAL_TIME
GBL_CPU_NNICE_UTIL	GBL_CPU_NORMAL_TIME_CUM
GBL_CPU_NNICE_UTIL_CUM	GBL_CPU_SYSCALL_UTIL
GBL_CPU_NNICE_UTIL_HIGH	GBL_CPU_SYSCALL_UTIL_CUM
GBL_CPU_NNICE_TIME	GBL_CPU_SYSCALL_UTIL_HIGH

GBL_CPU_NNICE_TIME_CUM	GBL_CPU_SYSCALL_TIME
GBL_CPU_REALTIME_UTIL	GBL_CPU_SYSCALL_TIME_CUM
GBL_CPU_REALTIME_UTIL_CUM	GBL_CPU_WAIT_UTIL
GBL_CPU_REALTIME_UTIL_HIGH	GBL_CPU_WAIT_TIME
GBL_CPU_REALTIME_TIME	GBL_CPU_WAIT_TIME_CUM
GBL_CPU_REALTIME_TIME_CUM	GBL_CPU_WAIT_UTIL_CUM
GBL_CPU_CSWITCH_UTIL	GBL_CPU_WAIT_UTIL_HIGH
Clase de métricas de aplicación	
APP_PRM_CPU_TOTAL_UTIL_CUM	APP_CPU_NORMAL_UTIL
APP_CPU_NNICE_UTIL	APP_CPU_USER_MODE_UTIL
APP_CPU_NNICE_TIME	APP_CPU_TOTAL_TIME
APP_CPU_TOTAL_UTIL	APP_CPU_SYS_MODE_TIME
APP_CPU_TOTAL_UTIL_CUM	APP_CPU_NICE_TIME
APP_CPU_SYS_MODE_UTIL	APP_CPU_REALTIME_TIME
APP_CPU_NICE_UTIL	APP_CPU_NORMAL_TIME
APP_CPU_REALTIME_UTIL	APP_CPU_USER_MODE_TIME
Clase de métrica PROC	
PROC_CPU_TOTAL_UTIL	PROC_CPU_REALTIME_UTIL
PROC_CPU_TOTAL_UTIL_CUM	PROC_CPU_REALTIME_UTIL_CUM
PROC_CPU_TOTAL_TIME	PROC_CPU_REALTIME_TIME
PROC_CPU_TOTAL_TIME_CUM	PROC_CPU_REALTIME_TIME_ CUM
PROC_CPU_SYS_MODE_UTIL	PROC_CPU_CSWITCH_UTIL
PROC_CPU_SYS_MODE_UTIL_CUM	PROC_CPU_CSWITCH_UTIL_CUM
PROC_CPU_SYS_MODE_TIME	PROC_CPU_CSWITCH_TIME

PROC_CPU_SYS_MODE_TIME_CUM	PROC_CPU_CSWITCH_TIME_CUM
PROC_CPU_USER_MODE_UTIL	PROC_CPU_INTERRUPT_UTIL
PROC_CPU_USER_MODE_UTIL_CUM	PROC_CPU_INTERRUPT_UTIL_CUM
PROC_CPU_USER_MODE_TIME	PROC_CPU_INTERRUPT_TIME
PROC_CPU_USER_MODE_TIME_CUM	PROC_CPU_INTERRUPT_TIME_CUM
PROC_CPU_NICE_UTIL	PROC_CPU_NORMAL_UTIL
PROC_CPU_NICE_UTIL_CUM	PROC_CPU_NORMAL_UTIL_CUM
PROC_CPU_NICE_TIME	PROC_CPU_NORMAL_TIME
PROC_CPU_NICE_TIME_CUM	PROC_CPU_NORMAL_TIME_CUM
PROC_CPU_NNICE_UTIL	PROC_CPU_SYSCALL_UTIL
PROC_CPU_NNICE_UTIL_CUM	PROC_CPU_SYSCALL_UTIL_CUM
PROC_CPU_NNICE_TIME	PROC_CPU_SYSCALL_TIME
PROC_CPU_NNICE_TIME_CUM	PROC_CPU_SYSCALL_TIME_CUM
PROC_CPU_ALIVE_TOTAL_UTIL	
PROC_CPU_ALIVE_USER_MODE_UTIL	
PROC_CPU_ALIVE_SYS_MODE_UTIL	
Clase de métrica BYCPU	
BYCPU_CPU_TOTAL_UTIL	BYCPU_CPU_INTERRUPT_TIME
BYCPU_CPU_TOTAL_UTIL_CUM	BYCPU_CPU_INTERRUPT_TIME_CUM
BYCPU_CPU_TRAP_TIME	BYCPU_CPU_INTERRUPT_UTIL
BYCPU_CPU_TRAP_TIME_CUM	BYCPU_CPU_INTERRUPT_UTIL_

	CUM
BYCPU_CPU_TRAP_UTIL	BYCPU_CPU_CSWITCH_TIME
BYCPU_CPU_TRAP_UTIL_CUM	BYCPU_CPU_CSWITCH_TIME_ CUM
BYCPU_CPU_USER_MODE_TIME	BYCPU_CPU_CSWITCH_UTIL
BYCPU_CPU_USER_MODE_TIME_ CUM	BYCPU_CPU_CSWITCH_UTIL_ CUM
BYCPU_CPU_USER_MODE_UTIL	BYCPU_CPU_VFAULT_TIME
BYCPU_CPU_USER_MODE_UTIL_ CUM	BYCPU_CPU_VFAULT_TIME_CUM
BYCPU_CPU_NICE_TIME	BYCPU_CPU_VFAULT_UTIL
BYCPU_CPU_NICE_TIME_CUM	BYCPU_CPU_VFAULT_UTIL_CUM
BYCPU_CPU_NICE_UTIL	BYCPU_CPU_REALTIME_TIME
BYCPU_CPU_NICE_UTIL_CUM	BYCPU_CPU_REALTIME_TIME_ CUM
BYCPU_CPU_NNICE_TIME	BYCPU_CPU_REALTIME_UTIL
BYCPU_CPU_NNICE_TIME_CUM	BYCPU_CPU_REALTIME_UTIL_ CUM
BYCPU_CPU_NNICE_UTIL	BYCPU_CPU_NORMAL_TIME
BYCPU_CPU_NNICE_UTIL_CUM	BYCPU_CPU_NORMAL_TIME_CUM
BYCPU_CPU_TOTAL_TIME	BYCPU_CPU_NORMAL_UTIL
BYCPU_CPU_TOTAL_TIME_CUM	BYCPU_CPU_NORMAL_UTIL_CUM
BYCPU_CPU_SYS_MODE_TIME	BYCPU_CPU_SYSCALL_TIME
BYCPU_CPU_SYS_MODE_TIME_ CUM	BYCPU_CPU_SYSCALL_TIME_ CUM
BYCPU_CPU_SYS_MODE_UTIL	BYCPU_CPU_SYSCALL_UTIL
BYCPU_CPU_SYS_MODE_UTIL_	BYCPU_CPU_SYSCALL_UTIL_

CUM	CUM
Clase de métrica SYSCALL	
SYSCALL_CPU_TOTAL_TIME	SYSCALL_CPU_TOTAL_TIME_CUM

cachemem

El parámetro `cachemem` del archivo **parm** permite configurar el agente para incluir el tamaño de la caché del búfer al generar informes de los datos de la utilización total de la memoria.

Puede establecer el parámetro `cachemem` en una de las siguientes opciones:

1. **Establezca el parámetro cachemem en free (f).**

   De manera predeterminada, el parámetro se establece en free (f). Se ven afectados los siguientes valores de métricas:

   GBL_MEM_UTIL - No incluye el tamaño de caché del búfer.

   GBL_MEM_FREE_UTIL - Incluye el tamaño de caché del búfer.
2. **Establezca el parámetro cachemem en user (u).**

   Se ven afectados los siguientes valores de métricas:

   GBL_MEM_UTIL - Incluye el tamaño de caché del búfer.

   GBL_MEM_FREE_UTIL - No incluye el tamaño de caché del búfer.

Nota:

En las máquinas AIX, debe establecer el parámetro cachemem en **free (f)** para que coincida con los comandos topas.

En las máquinas Solaris, el parámetro cachemem solo se aplica a la Caché de reemplazo adaptativo (ARC) para ZFS.

De manera predeterminada, el parámetro `cachemem` se establece en **user (u)**. Puede establecer este parámetro en **free (f)** en el archivo parm y después reiniciar el recopilador de datos. GBL_MEM_UTIL excluye la caché ZFS ARC cuando el parámetro cachemem del archivo parm se establece en **free**.

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.

Nota: 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.

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

Nombre de aplicación

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

Cuando especifica una aplicación, se aplican las reglas o convenciones siguientes:

- 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 nombre `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`.
- Puede definir hasta 998 aplicaciones.
- Se puede tener un máximo de 4096 especificaciones `file`, `user`, `group`, `argv1` y `cmd` en todas las aplicaciones combinadas.
- 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
```

Donde:

xyz* cuenta como una única especificación aunque pueda coincidir con más de un archivo de programa.

Nota: 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 de aplicación enviada. Se genera un error si no se encuentra ninguna instrucción `application`.

El nombre file name puede ser uno de los siguientes:

- **En UNIX/Linux:**
  - 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.
- **En Windows**:
  - Un único archivo de programa UNIX como `winword`.
  - Un grupo de archivos de programa (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 hacia el máximo de 1000 para todas las especificaciones de archivos.

Nota: En Windows, cuando define los archivos ejecutables para una aplicación en el archivo `parm`, no se requieren extensiones de archivo. Por ejemplo, puede definir winword en el archivo `parm` sin su extensión `.exe`.

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:

En UNIX/Linux:

```
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
```

En Windows:

```
application payroll
file account1,basepay,endreport
application Office
file winword* excel*
file 123* msaccess*
```

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

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

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` es **True**, si es el nombre `class` o `jar` de los procesos Java.

El parámetro `argv1` 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
```

El ejemplo siguiente 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. La cadena del comando consta del programa ejecutado y sus argumentos (parámetros). 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`. Solo se puede tener una selección por línea; sin embargo, puede tener varias líneas.

Los ejemplos siguientes muestran cómo se utiliza el 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, 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 se especifica ningún 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 los nombres de grupo de usuarios que 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
```

La prioridad de los procesos oscila entre -511 y 255, dependiendo de la plataforma de HPE Operations Agent 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.

Nota: 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*
o
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
/opt/local/bin/appserver -xyz -option1	**xyz**	firstthreesvrs
./appserver -option5	**root**	othersvrs
./appserver -xyz -option2 -abc	**root**	firstthreesvrs
./appsvr -xyz -option2 -abc	**xyz**	othersvrs
./appclient -abc	**root**	other
./appserver -mno -option4	**xyz**	oursvrs
appserver -option3 -jkl	**xyz**	firstthreesvrs
/tmp/bleh -xyz -option1	**xyz**	othersvrs

Configuración de intervalos de registro de datos

Los intervalos de recopilación predeterminados utilizados por **oacore** 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 collection interval 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 de los datos de proceso deben ser un múltiplo del intervalo de subproc (consulte subprocinterval), 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.

Configuración de recopilación de datos para marcos

Puede recopilar datos de rendimiento desde todos los LPAR de AIX disponibles en un único marco mediante la instalación de HPE Operations Agent en solo un nodo LPAR. También puede configurar al agente para que recopile datos de rendimiento desde el marco AIX donde residen todos los LPAR monitorizados.

Puede tener las ventajas siguientes si habilita la monitorización de los marcos:

- Recopile la información de configuración:
  - Nombre y UUID del marco donde residen los LPAR monitorizados
  - Modelo, número de serie y tipo del marco
  - Configuración de CPU y capacidad de memoria del marco
- Con la información adicional, puede analizar la utilización de recursos del marco
- Utilice las herramientas de análisis de datos (como HP Performance Manager) para analizar la relación de consumo de CPU del marco y todos los LPAR.

Tarea 1: Configurar el acceso SSH sin contraseña

En el sistema de la consola de gestión de hardware (HMC), puede configurar el acceso SSH sin contraseña entre el nodo de LPAR (nodo donde ha instalado el agente) y el sistema HMC. Después de configurar el acceso a un SSH sin contraseña para un usuario, puede ejecutar los comandos mediante el protocolo SSH en el sistema HMC de manera remota desde el nodo LPAR sin tener que proporcionar una contraseña para la autenticación.

Siga estos pasos:

Configurar el acceso SSH sin contraseña

En el nodo LPAR

1. Inicie sesión en el servidor nodo LPAR como usuario raíz.
2. Ejecute el comando para crear un directorio:

   ```
   # mkdir .ssh
   ```

   Nota: Puede ejecutar el comando siguiente para suprimir un archivo `.ssh #`

```
rm -rf .ssh
```

3. Ejecute el comando para generar un par de claves rsa pública/privada:

   ```
   # ssh-keygen -t rsa
   ```

4. Especifique un archivo para guardar la clave:

   ```
   //.ssh/<nombre_de_archivo>_rsa
   ```

   La clave primava se guarda en `//.ssh/ <filename>_rsa` y la clave pública se guarda en `//.ssh/ <filename>_rsa.pub.`

5. Ejecute los siguientes comandos para ver la clave:

   ```
   # cat ./ssh
   ```

   ```
   # cat .ssh/<nombre_de_archiv>_rsa.pub
   ```

Se muestra la clave generada.

En la máquina HMC

1. Inicie sesión en la máquina HMC con las credenciales de usuario.
2. Ejecute el comando siguiente para establecer el acceso sin contraseña:

   ```
   mkauthkeys -a 'ssh-rsa<key>’
   ```

   En esta instancia <key> es la clave generada en el paso 3.

   Nota: La clave completa debe mencionarse en una única línea.

Comprobación del acceso SSH sin contraseña

Siga estos pasos para comprobar si se crea el acceso SSH sin contraseña para el usuario:

1. Inicie sesión en el servidor nodo LPAR como usuario raíz.
2. Ejecute el comando siguiente:

   ```
   ssh hmcuser@hmchost.example.domain.com lssyscfg r sys
   ```

Si el comando muestra la lista de marcos sin solicitar una contraseña, el usuario lo ha configurado correctamente y puede monitorizar la utilización de los marcos.

Tarea 2: Habilitar la monitorización de la utilización de los marcos en el sistema HMC

1. Inicie sesión en el sistema HMC como usuario raíz.
2. Ejecute el comando siguiente:

```
chlparutil -r config -s 60
```

Tarea 3: Configurar HPE Operations Agent

1. Inicie sesión en el nodo LPAR.
2. Vaya al directorio `/var/opt/perf`.
3. Cree un nuevo archivo.
4. Abra el archivo con un editor de texto y después agregue el contenido siguiente al archivo:

   <hmcusername>@<hmc_fqdn>

   En este ejemplo:

   <hmcusername> es el usuario al que se le concedió el acceso SSH sin contraseña al sistema HMC en "Tarea 1: Configurar el acceso SSH sin contraseña" en la página 196.

   <hmc_fqdn> es el nombre de dominio completo del host de HMC.

   Por ejemplo:

   ```
   hmcusername@hmchost.example.domain.com
   ```

5. Guarde el archivo como `hmc` en el directorio `/var/opt/perf`.
6. Configure el parámetro `logicalsystem` del archivo `parm` e inicie el proceso de recopilación. Para obtener más información sobre la configuración del archivo `parm`, consulte "Uso del archivo parm" en la página 161.
7. Reinicie Componente Performance Collection.

   HPE Operations Agent recopila los datos de rendimiento específicos del marco a intervalos de cinco minutos.

Habilitación de las métricas de llamada del sistema de nivel de proceso y global para GlancePlus en Linux

En una máquina Linux, puede habilitar la recopilación de valores de métricas de rendimiento de GBL_SYSCALL, SYSCALL y PROCSYSCALL para GlancePlus. Antes de comenzar a configurar la recopilación de métricas, asegúrese de detener todos los procesos que usan FTRACE en su máquina Linux.

Nota: Las métricas se admiten para RHEL/OEL/CentOS 6.1 y versiones superiores, SLES 11 SP1 y versiones superiores, Ubuntu 11.10 y versiones

superiores, Debian 6.0 y versiones superiores.

Puede habilitar la recopilación de métricas para GlancePlus mediante *uno* de los métodos siguientes:

- Configuración de la recopilación de métricas mediante `init_ftrace.sh`
- Configuración de la recopilación de métricas mediante pasos manuales

Configuración de la recopilación de métricas mediante **init_ftrace.sh**

Para configurar la recopilación de métricas en la máquina Linux mediante `init_ftrace.sh`, siga estos pasos:

1. Inicie sesión en el nodo con los privilegios necesarios.
2. Para detener todos los procesos del componente Performance Collection en la máquina Linux, ejecute el siguiente comando:

   ```
   /opt/perf/bin/ovpa stop all
   ```
3. Detenga todos los procesos `midaemon` en su máquina Linux. Para comprobar el proceso `midaemon`, ejecute el comando siguiente:

   ```
   ps -ef I grep midaemon
   ```

   Si el proceso `midaemon` se está ejecutando en su máquina Linux, ejecute el comando siguiente:

   ```
   killall midaemon
   ```

   Esto detiene todos los procesos `midaemon`.
4. Para configurar la recopilación de métricas, ejecute el siguiente comando:

   ```
   /opt/perf/bin/init_ftrace.sh
   ```

   El comando monta debugfs y habilita FTRACE en la máquina Linux. Si FTRACE ya se está ejecutando en la máquina con otras aplicaciones aparecerá el siguiente mensaje indicando

   si desea restablecer la interfaz de FTRACE para usarla con midaemon. (Y/N).

   Puede seleccionar *una* de las siguientes opciones:

   Y – Restablece la interfaz FTRACE en su máquina Linux.

   N – No configura la recopilación de métricas. No puede ver las métricas en GlancePlus.
5. Para iniciar todos los procesos del componente Performance Collection junto con el proceso `midaemon` y la colección de métricas, ejecute el comando

siguiente:

```
/opt/perf/bin/ovpa start all
```

Utilice las pantallas siguientes de GlancePlus para comprobar las métricas de rendimiento.

Y - para comprobar las métricas SYSCALL

L - para comprobar las métricas PROCSYSCALL

Nota: Cuando GlancePlus no está en la pantalla de llamada del sistema, el rendimiento del sistema no se ve afectado aunque debugfs se monte y FTRACE esté habilitado. Para desmontar debugfs de **/sys/kernel/debug**, puede ejecutar el comando `umount /sys/kernel/debug`.

Configuración de la recopilación de métricas mediante pasos manuales

Para configurar la recopilación de métricas manualmente en la máquina Linux, siga estos pasos:

1. Inicie sesión en el nodo con los privilegios necesarios.
2. Para detener todos los procesos del componente Performance Collection en la máquina Linux, ejecute el siguiente comando:

   ```
   /opt/perf/bin/ovpa stop all
   ```
3. Detenga todos los procesos `midaemon` en su máquina Linux. Para comprobar el proceso `midaemon`, ejecute el comando siguiente:

   ```
   ps -ef I grep midaemon
   ```

   Si el proceso `midaemon` se está ejecutando en su máquina Linux, ejecute el comando siguiente:

   ```
   killall midaemon
   ```

   Esto detiene todos los procesos `midaemon`.
4. Montar debugfs manualmente en la máquina Linux

   Nota: Omita este paso si debugfs ya está montado en la máquina Linux. Para comprobar si debugfs ya está montado, ejecute el comando siguiente:

   ```
   cat /proc/mounts |grep debugfs
   ```

   El comando genera *una* de las salidas siguientes:

nodev /sys/kernel/debug debugfs rw,relatime 0 0 - debugfs ya está montado en la máquina.

nil – debugfs no está montado en la máquina.

Para montar debugfs de forma manual, siga estos pasos:

i. Ejecute el comando siguiente:

```
mount -t debugfs nodev /sys/kernel/debug
```

La salida aparece como

nil – debugfs está correctamente montado en la máquina.

Si debugfs no se ha montado correctamente en la máquina, se muestra un mensaje de error.

5. Habilitar FTRACE manualmente en la máquina Linux

Nota: Omita este paso si FTRACE ya está montado en la máquina Linux. Para comprobar si FTRACE ya está montado, ejecute el comando siguiente:

```
cat /proc/sys/kernel/ftrace_enabled
```

El comando genera *una* de las salidas siguientes:

1 – FTRACE está correctamente montado en la máquina.

0 – FTRACE no está correctamente montado en la máquina.

Para habilitar FTRACE de forma manual, siga estos pasos:

i. Ejecute el comando siguiente:

```
echo "1" >/proc/sys/kernel/ftrace_enabled
```

La salida aparece como

nil – FTRACE está correctamente montado en la máquina.

Si FTRACE no se ha montado correctamente en la máquina, se muestra un mensaje de error.

6. Para iniciar todos los procesos del componente Performance Collection junto con el proceso `midaemon` y la colección de métricas, ejecute el comando siguiente:

```
/opt/perf/bin/ovpa start all
```

Utilice las pantallas siguientes de GlancePlus para comprobar las métricas de rendimiento.

Y - para comprobar las métricas SYSCALL

L - para comprobar las métricas PROCSYSCALL

Nota: Cuando GlancePlus no está en la pantalla de llamada del sistema, el rendimiento del sistema no se ve afectado aunque debugfs se monte y FTRACE esté habilitado. Para desmontar debugfs de **/sys/kernel/debug**, puede ejecutar el comando `umount /sys/kernel/debug`.

Solución de problemas

Esta sección describe las soluciones y medidas a tomar para resolver los problemas más comunes que pueden surgir al configurar la recopilación de métricas.

Problema:

Si se produce un error en el proceso `midaemon`, consulte `/var/opt/perf/status.mi` para obtener más información. Puede experimentar un error de proceso `midaemon` si se ha quedado sin espacio en su segmento de memoria compartida.

Solución:

Para evitar este problema, tiene que borrar la memoria compartida no deseada. Para borrar la memoria compartida, siga estos pasos:

1. Ejecute el comando siguiente:

   ```
   ipcs -m | grep 0x0c6629c9
   ```

   El comando generará una salida con un valor variable en el segundo campo de datos. Para borrar la memoria compartida, ejecute el comando siguiente:

   ```
   ipcrm –m <valor_de_campo>
   ```

   Ejemplo

   El ejemplo siguiente muestra cómo borrar la memoria compartida:

   ```
   ipcs -m | grep 0x0c6629c9
   output= 0x0c6629c9 18841617 root 640 8704448 7
   ipcrm –m 18841617
   ```

2. Para reiniciar `midaemon`, ejecute el comando siguiente:

   ```
   /opt/perf/bin/ovpa restart
   ```

Normalización de métricas de CPU en sistemas habilitados para hipersubprocesamientos o 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, Componente Performance Collection 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 HPE 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 Componente Performance Collection para registrar todas las métricas relacionadas con la CPU con normalización basada en núcleo. En otras plataformas, se puede configurar Componente Performance Collection para calcular las métricas relacionadas con la CPU de la clase GLOBAL con la normalización basada en núcleo antes del registro.

Para configurar Componente Performance Collection 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**.

   Nota: 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:

   `/opt/perf/bin/midaemon –ignore_mt`
5. Inicie HPE Operations Agent ejecutando el siguiente comando:

   `/opt/OV/bin/opcagt –start`

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

Si reinicia HPE Operations Agent, Componente Performance Collection empieza a registrar de nuevo los datos de la CPU con la normalización basada en subprocesos y debe volver a configurar Componente Performance Collection usando los pasos anteriores. Para habilitar el agente para que utilice siempre la normalización basada en núcleo, siga estos pasos:

1. En el nodo de agente, vaya a la ubicación siguiente:

   /var/opt/perf
2. Abra el siguiente archivo con un editor de texto:

   vppa.env
3. En el parámetro MIPARMS, seleccione ignore_mt.
4. Guarde el archivo.
5. Reinicie el agente ejecutando los siguientes comandos:

   `/opt/OV/bin/opcagt –start`

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 HPE Operations Agent usando el comando siguiente:

   En Windows:

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

   En Linux y Solaris:

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

   En AIX:

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

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

Capítulo 6: Uso del programa utility

El programa `utility` es una herramienta que permite administrar y notificar información sobre el archivo de parámetros de recopilación (`parm`) y el archivo de definiciones de alarmas (`alarmdef`). El programa `utility` puede utilizarse para realizar las tareas siguientes:

- Analizar los almacenes de datos de HPE Operations Agent y generar un informe que muestre:
  - Cobertura de fechas y horas
  - Efectos de la configuración del proceso y aplicación en el archivo de parámetros de recopilación (**parm**)
- Comprobar si el archivo `parm` contiene errores o advertencias de sintaxis
- Comprobar si el archivo `alarmdef` contiene errores o advertencias de sintaxis
- Procesar los datos almacenados en el almacén de datos con definiciones de alarmas para detectar condiciones de alarma en datos históricos

Ejecución del programa utility

HPE Operations Agent 12.xx solo admite el modo de línea de comandos para ejecutar el programa `utility`. Las opciones de comandos y sus argumentos asociados se pasan al programa `utility` a través de la interfaz de línea de comandos. La interfaz de línea de comandos funciona en todas las plataformas y permite que el programa `utility` pueda invocarse fácilmente por scripts de shell posibilitando que tanto las entradas como las salidas sean redireccionadas a canalizaciones se redireccionen.

Los argumentos y opciones de línea de comandos se enuncian en la tabla siguiente:

Tabla 1: Argumentos de línea de comandos

Opciones de comando	Argumento	Descripción
`-f`	`filename`	Especifica el nombre de archivo de salida. **Por ejemplo:**

Opciones de comando	Argumento	Descripción
		Para especificar `utilrept` como el archivo de salida para todos los informes de **utility**, ejecute el comando siguiente: `utility -f utilrept -d -xs` Para obtener más información, consulte filename.
`-D`		Habilita los detalles para la comprobación de los archivos `analyze`, `scan` y `parm`. Para obtener más información, consulte Comando detail.
`-d`		Deshabilite los detalles para la comprobación de archivos `analyze` y `parm`. Para obtener más información, consulte Comando detail.
`-T`		Genera formatos de informes de salida terse.
`-v`		Muestra los comandos cuando se ejecutan en la línea de comandos.
`-xp`	`parmfile`	Comprueba la sintaxis

Opciones de comando	Argumento	Descripción
		de un archivo `parm`. **Por ejemplo:** Para comprobar los detalles del archivo `parm`, ejecute el comando siguiente: `utility -xp` Para obtener mayor información, consulte Comando parmfile.
`-xc`	`alarmdef`	Compruebe la sintaxis y establezca el nombre del archivo `alarmdef` para su uso con `-xa` (o el comando `analyze`). **Por ejemplo:** Para obtener mayor información, consulte Comando checkdef.
`-xa`		Analiza los datos almacenados en el almacén de datos comparándolos con el archivo `alarmdef`. **Por ejemplo:** Para ver los eventos de alarma así como el resumen de alarma, ejecute el comando siguiente: `utility -xa -D` Para obtener mayor

Opciones de comando	Argumento	Descripción
		información, consulte Comando analyze.
`-xs`	`datastore`	Explora el almacén de datos y genera un informe. **Por ejemplo:** Para examinar los datos, ejecute el comando siguiente: `utility -D -xs` Para obtener mayor información, consulte Comando scan.
`-? o ?`		Muestra la sintaxis de la línea de comandos.

Informe de exploración del programa utility

El comando `scan` del programa `utility` lee el almacén de datos y escribe un informe sobre su contenido. El informe de exploración del programa utility proporciona la información sobre el espacio de disco utilizado por cada clase de datos.

Por ejemplo:

Si ejecuta el comando `utility -xs`, se generan el informe de resumen de clase y el informe de exploración del programa utility, como se muestra en la figura siguiente:

```
                # utility -xs
      CLASSNAME                        RECORDS     STARTTIME           ENDTIME           HH:MM:SS
-------------------------------------------------------------------------------------------------
SCOPE::CORE                                  0 0000/00/00 00:00:00  0000/00/00 00:00:00  00:00:00
SCOPE::FILESYSTEM                       209150 2015/07/13 15:13:43  2015/07/21 16:01:00  192:47:17
SCOPE::CPU                               41830 2015/07/13 15:13:43  2015/07/21 16:01:00  192:47:17
SCOPE::GLOBAL                            20915 2015/07/13 15:13:43  2015/07/21 16:01:00  192:47:17
SCOPE::APPLICATION                       45116 2015/07/13 15:13:43  2015/07/21 16:01:00  192:47:17
SCOPE::PROCESS                          106857 2015/07/13 15:13:43  2015/07/21 16:01:00  192:47:17
SCOPE::DISK                              44536 2015/07/13 15:13:43  2015/07/21 16:01:00  192:47:17
SCOPE::NETIF                             41830 2015/07/13 15:13:43  2015/07/21 16:01:00  192:47:17
SCOPE::LVOLUME                               0 0000/00/00 00:00:00  0000/00/00 00:00:00  00:00:00
SCOPE::TRANSACTION                           0 0000/00/00 00:00:00  0000/00/00 00:00:00  00:00:00
SISPI::LOGINS                             1690 2015/07/14 16:00:02  2015/07/21 16:00:02  168:00:00
SISPI::GLOBAL                              169 2015/07/14 16:00:03  2015/07/21 16:00:03  168:00:00
```

Si ejecuta el comando `utility -xs -D`, verá la salida siguiente:

La comparación entre los informes de exploración (versión 11.xx y 12.xx) se indica en la sección Informe de exploración de programa utility en el capítulo *Comparación de HPE Operations Agent 12.xx con versiones anteriores*.

Comandos del programa utility

Este capítulo ofrece una descripción detallada de los comandos del programa **utility**.

analyze

El comando `analyze` permite analizar los datos contenidos en el almacén de datos en función de 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.

El archivo de definiciones de alarma predeterminado para el modo de línea de comandos es **/var/opt/perf/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.

El usuario podrá ejecutar de manera opcional los comandos `start`, `stop` y `detail` con `analyze` para personalizar el proceso de análisis.

Para sólo ver el informe de resumen de alarmas, debe emitirse el comando `detail off`. En el modo de línea de comandos, `detail off` es el valor predeterminado. Para ver los eventos de alarma así como el resumen de alarmas, especifique el comando `detail on (-D)`.

Mientras se ejecuta el comando `analyze`, éste indicará eventos de alarma tales como `alarm start`, `end` y `repeat status`. 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`.

Por ejemplo

Para ver el informe de resumen de alarmas de rendimiento, ejecute el siguiente comando:

```
utility -xa
```

Verá la salida siguiente:

```
Performance Alarm Summary:

Alarm   Count   Minutes
    1       3        25
    3       4        20
    4       16      125

Analysis coverage using "/var/opt/perf/alarmdef":

Start: 05/05/2014 00:00:00    Stop: 05/05/2014 16:55:00

Total time analyzed:  Days: 0  Hours: 16  Minutes: 55
```

Para ver los eventos de alarma así como el resumen de alarma, ejecute el comando siguiente:

```
utility -xa -D
```

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.

Por 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 -xc
```

Cuando esté seguro de que las definiciones de alarma son correctas, se podrán procesar contrastándolas con los datos contenidos en el almacén de datos usando el comando `analyze`.

detail

El comando `detail` permite controlar el nivel de detalles imprimidos en los informes `analyze`, `parmfile` y `scan`. El valor predeterminado es `detail off` en el modo de línea de comandos.

Parámetros

detail	`on`	-D	Imprime el contenido del archivo **parm** así como los errores del archivo **parm**. Imprime informes completos de analyze y scan.
	`off`	-d	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 collector, 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.

Para obtener ejemplos del uso del comando detail, consulte las descripciones de los comandos analyze, parmfile y scan que encontrará en este capítulo.

help

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

Por ejemplo

Para ver los argumentos de la línea de comandos del programa Utility, ejecute el siguiente comando:

```
utility -?
```

filename

El comando `filename` 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 `filename`.

Por ejemplo

```
utility -f <nombre de archivo> -d -xs
```

Para especificar `utilrept` como el archivo de salida para todos los informes de **utility**, ejecute el comando siguiente:

```
utility -f utilrept -d -xs
```

El parámetro `<nombre de archivo>` del comando `filename` debe representar un nombre de archivo válido al cual el usuario tiene acceso de escritura. Salida anexada al final del archivo existente. Si el archivo no existe, se creará.

Para determinar el archivo de salida actual, ejecuta el comando `filename` sin parámetros:

Si el archivo de salida no es `standard output`, la mayoría de los comandos se reproducirán en el archivo de salida conforme se introduzcan.

Nota: En los sistemas AIX, no puede usar la opción `-f` con el comando `utility -xs` para especificar el archivo de salida.

parmfile

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

Por ejemplo

Para comprobar la sintaxis del archivo **parm**, haga clic en el comando siguiente.

```
utility -xp
```

Verá la salida siguiente:

```
33 file  names used to define applications
1 user  names used to define applications
0 group names used to define applications

Parm File: "/var/opt/perf/parm" had 0 Warnings.
```

El comando `parmfile` comprueba la sintaxis del archivo **parm** actual y notifica toda advertencia o error.

Para ver la lista de configuración de parámetros de registro, ejecute el comando siguiente:

```
utility -xp -v
```

scan

El comando `scan` permite leer el almacén de datos y escribir un informe sobre su contenido. Los comandos siguientes afectan la operación de la función de exploración:

`detail`	Especifica la cantidad de detalles del informe. El valor predeterminado, `detail on`, especifica totalidad de detalles.
`list`	Redirecciona la salida a otro archivo. El predeterminado es incluir en dispositivo de lista estándar.

Para obtener más información, consulte detail y list.

Por ejemplo: Para examinar los datos, ejecute el comando siguiente:

```
utility -D -xs
```

Capítulo 7: Uso del programa extract

El programa `Extract` se usa para recuperar y analizar los datos históricos registrados en el almacén de datos de HPE Operations Agent. El programa `Extract` realiza la función de exportación. Lee los datos del almacén de datos y exporta los resultados a los archivos de salida en formato ASCII.

Nota: El proceso `oacore` debe ejecutarse para que funcione el programa `extract`.

Ejecución del programa Extract con la interfaz de línea de comandos

HPE Operations Agent 12.xx solo admite el modo de línea de comandos para ejecutar el programa Extract. Para recuperar los datos del almacén de datos, ejecute el programa Extract usando el modo de línea de comandos. 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. Funciona bien en entornos UNIX convencionales ya que permite que los scripts de shell puedan invocar el programa `Extract` con facilidad y que tanto las entradas como las salidas se redireccionen a canalizaciones UNIX.

Nota: Durante una actualización de HPE Operations Agent versión 11.xx a 12.xx, los datos más antiguos - los datos almacenados en los archivos de base de datos CODA, archivos de registro `scope` y archivos de registro DSI se conservan en modo de solo lectura. El programa extract puede leer los datos del almacén de datos antiguo y nuevo.

La siguiente tabla presenta las opciones y argumentos de línea de comandos.

Tabla 3: Argumentos de línea de comandos

Opción de comando	Argumento		Descripción
`-b`	`date`	`time`	Especifica la fecha y la hora de inicio de una función `export`. Sintaxis: mm/dd/aaaa hh:mm:ss
`-B`		`UNIX`*`hora`*	Especifica la hora de inicio en

Tabla 3: Argumentos de línea de comandos, continuación

Opción de comando	Argumento		Descripción
		de inicio	formato UNIX de una función `export`.
`-e`	`date`	`time`	Especifica la fecha y la hora de finalización de una función `export`. Sintaxis: mm/dd/aaaa hh:mm:ss
`-E`		`UNIX`*hora de finalización*	Especifica la hora de finalización en formato UNIX para una función `export`.
`-l`	`logfile`		Especifica el archivo de registro de entrada para los datos DSI.
`-C`	`classname`		Especifica los datos autodescriptivos (DSI) con la función `export`.
`gapcdznituyhx` `GADZNITUYHX`			Especifica el tipo de archivo con el que se usará la función `export`: `g` = detalles globales `a` = detalles de aplicación `p` = detalles de proceso `c` = detalles de configuración `d` = detalles de dispositivo de disco `z` = detalles de volumen lógico `n` = detalles de netif `i` = detalles de sistemas lógicos

Tabla 3: Argumentos de línea de comandos, continuación

Opción de comando	Argumento		Descripción
			Nota: las plataformas Windows y Linux no admiten detalles de los sistemas lógicos. `t` = detalles de transacción `u` = detalles de CPU `y` = detalles de sistema de archivos `h` = detalles del adaptador de bus del host (HBA) `x` = detalle básico `G` = resumen global `A` = resumen de aplicación `D` = resumen de dispositivo de disco `Z` = resumen de volumen lógico `N` = resumen de netif `I` = resumen de sistemas lógicos **Nota:** las plataformas Windows y Linux no admiten el resumen de los sistemas lógicos. `T` = resumen de transacción `U` = resumen de CPU `Y` = resumen de sistema de

Tabla 3: Argumentos de línea de comandos, continuación

Opción de comando	Argumento		Descripción
			archivos `H` = resumen de HBA `x` = resumen básico
`-r`	`export template file`		Especifica un archivo de plantilla de exportación para la función `export` **Nota:** Si la ruta del archivo de plantilla de exportación contiene espacios, asegúrese de que los rodea de comillas dobles.
`-f`	`filename`		Especifica el nombre de archivo de salida.
`-xp`	`xopt`		Exporta datos a archivos de formato ASCII.
`-? o ?`			Muestra la sintaxis de la línea de comandos.

Uso de la función de exportación

El comando de exportación lee los almacenes de datos de la base de datos y exporta los resultados en los archivos de salida. Los archivos de salida 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

Siga estos pasos:

1. Especifique el tipo de clase o el tipo de datos. Consulte "Clases de métricas de rendimiento personalizadas" en la página siguiente.

   Por ejemplo: `-a` (clase de aplicación), `-g` (clase global)

2. Especifique el archivo de salida para los datos de exportación.

   Nota:

   Si no especifica el nombre del archivo de salida, los datos exportados se escriben en el archivo de salida predeterminado. El archivo de salida predeterminado depende de la clase de datos que se exporta.

   Por ejemplo: Para exportar datos detallados de la clase global se usa el archivo de salida predeterminado `xfrdGLOBAL.asc`. Consulte "Tabla 5: Archivos de salida predeterminados" en la página 221.

3. Ejecute el comando de exportación.
4. Consulte los datos exportados. Consulte "Salida de datos exportados" en la página 226.

Nota: Se proporcionan los archivos de plantilla de exportación `reptfile` y `reptall` con Componente Performance Collection. Estos archivos se encuentran en el directorio `/var/opt/perf/`. `Reptfile` tiene un conjunto predefinido de métricas cuyas marcas de comentario se quitan. Se usa de manera predeterminada para realizar tareas de exportación comunes. En el archivo `reptall` todas las métricas se comentan y se usan para exportar métricas específicas. Para personalizar el archivo de plantilla de exportación, consulte Producción de un archivo de salida personalizado.

Por ejemplo:

Para exportar los datos detallados de la CPU desde el 26 de julio de 2014, escriba el siguiente comando:

```
extract -u -b 7/26/14 -xp
```

Donde

`-u`	Especifica los detalle de CPU.
`-b`	Especifica la fecha de inicio de la función de exportación.
`-xp`	Exporta datos a archivos de formato externo

Puesto que no se ha especificado ningún archivo de plantilla de exportación, se usa el predeterminado: `reptfile`.

Clases de métricas de rendimiento personalizadas

Es posible exportar las siguientes clases de datos:

Tabla 4: Tipos de clases

Tipo de clase	Intervalo de resumen
global	5, 15, 30, 60, 180, 360, 720, 1440, 10080 minutos.
application	5, 15, 30, 60, 180, 360, 720, 1440, 10080 minutos.
process	Esta clase de métricas no está resumida
disk device	5, 15, 30, 60, 180, 360, 720, 1440, 10080 minutos.
lvolume	5, 15, 30, 60, 180, 360, 720, 1440, 10080 minutos.
transaction	5, 15, 30, 60, 180, 360, 720, 1440, 10080 minutos.
configuration	5, 15, 30, 60, 180, 360, 720, 1440, 10080 minutos.
netif	5, 15, 30, 60, 180, 360, 720, 1440, 10080 minutos.
cpu	5, 15, 30, 60, 180, 360, 720, 1440, 10080 minutos.
filesystem	5, 15, 30, 60, 180, 360, 720, 1440, 10080 minutos.
host bus adapter	5, 15, 30, 60, 180, 360, 720, 1440, 10080 minutos.
core	5, 15, 30, 60, 180, 360, 720, 1440, 10080 minutos.

Nota: Si menciona otros intervalos de resumen, aparte de los mencionados en la tabla, el programa de extracción se resume al intervalo de resumen más próximo.

Archivos de salida

Puede usar los archivos de salida predeterminados para exportar datos o especificar un archivo de salida antes de enviar el comando `export`.

- Si especifica el archivLa ventana Trace Properties presenta los siguientes deto de salida antes de enviar el comando `export`, todos los datos se exportan a este único archivo.

  Por ejemplo:

  Para exportar los datos detallados globales al archivo de salida denominado `myout`, ejecute el comando siguiente:

  ```
  extract -g -f myout -xp
  ```

Donde

`-g`	Especifica los detalles globales.
`-f`	Especifica el nombre de archivo de salida.
`myout`	Es el nombre del archivo de perfil.
`-xp`	Exporta datos.

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

  Por ejemplo:

  ```
  extract -xp -g
  ```

  El comando de exportación causa que la clase global de datos se exporte al archivo `xfrdGLOBAL.asc`.

La tabla siguiente enumera todos los archivos de salida predeterminados.

Tabla 5: Archivos de salida predeterminados

Archivos de salida predeterminados	Descripción
`xfrdGLOBAL.asc`	Archivo de datos de detalle globales
`xfrsGLOBAL.asc`	Archivo de datos de resumen global por horas
`xfrdAPPLICATION.asc`	Archivo de datos de detalle de aplicación
`xfrsAPPLICATION.asc`	Archivo de datos de resumen de aplicación por horas
`xfrdPROCESS.asc`	Archivo de datos de detalle de proceso
`xfrdDISK.asc`	Archivo de datos de detalle de dispositivo de disco
`xfrsDISK.asc`	Archivo de datos de resumen de dispositivo de disco por horas
`xfrdVOLUME.asc`	Archivo de datos de detalle del volumen lógico
`xfrsVOLUME.asc`	Archivo de datos de resumen de volumen lógico

Tabla 5: Archivos de salida predeterminados, continuación

xfrdNETIF.asc	Archivo de datos de detalle de NETIF
xfrsNETIF.asc	Archivo de datos de detalle de resumen de netif
xfrdCPU.asc	Archivos de datos de detalle de CPU
xfrsCPU.asc	Archivo de datos de resumen de CPU
xfrdFILESYSTEM.asc	Archivo de datos de detalle de sistema de archivos
xfrsFILESYSTEM.asc	Archivo de datos de resumen de sistema de archivos
xfrdTRANSACTION.asc	Archivo de datos de detalle de transacción
xfrsTRANSACTION.asc	Archivo de datos de resumen de transacción
xfrdCONFIGURATION.asc	Archivo de datos de configuración
xfrdHBA.asc	Archivo de datos de detalle del adaptador de bus de host (HBA)
xfrsHBA.asc	Archivo de datos de resumen de HBA
xfrdCORE.asc	Archivo de datos de detalle principal
xfrsCORE.asc	Archivo de datos de resumen principal

donde `ext= asc` (ASCII).

Nota: 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.

Los nombres de archivo predeterminados se crean a partir del nombre del tipo de datos. El prefijo es `xfrd` o `xfrs` dependiendo de los datos. `xfrd` se usa para los datos de detalle y `xfrs` se usa para los datos de resumen.

Por ejemplo, classname = ACCTG_INFO puede tener los nombres de archivo de **export** de:

xfrdACCTG_INFO.asc	datos ASCII detallados para ACCT_INFO
xfrsACCTG_INFO.asc	datos ASCII resumidos para ACCT_INFO

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:

```
HEADINGS    [ON]
            [OFF]
SEPARATOR= "|"
SUMMARY=VALUE
OUTPUT=Filename
DATA TYPE DATATYPE
METRICS
```

Por ejemplo:

```
HEADINGS ON
SEPARATOR="|"
SUMMARY=60

*************************************************************************
**
** The rest of the file consists of specifics about the metrics to be
** exported for a single data type. Each exportable class may have one
** set of definitions in which the following lines appear.
**
** DATA TYPE specifies the type (class) of data. (required)
**
** OUTPUT    specifies the name of the output file where the exported data is
**           to be written.  OUTPUT may be specified for each class named.
**           (optional)
**
**
**  Individual metric names which belong to the class. Metrics will be exported
**  in the order listed whenever possible. In order to be exported, metrics
**  must not be commented out (that is, they must not contain an asterisk before
**  the metric name).
**
** This sample report was generated with a section for each data type which
** is available in the currently open log file. Following the DATA TYPE
** line, each metric which is available for export for this data type is
** listed, but commented out. To select metrics for your report, delete
** the asterisk (*) in the first column.
**
*************************************************************************
*************************************************************************

*************************************************************************
DATA TYPE GLOBAL

**............................. Global Record Identification Metrics

* GBL_PROC_SAMPLE
* GBL_SYSTEM_UPTIME_HOURS
* GBL_SYSTEM_UPTIME_SECONDS
* GBL_STATTIME
* GBL_INTERVAL
* GBL_CSWITCH_RATE
* GBL_INTERRUPT_RATE
* GBL_INTERRUPT
```

Parámetros

headings	Especifica si se incluyen o no los encabezados de columna para las métricas enumeradas en el archivo `export`. Si se especifica `headings off`, no se escriben encabezados de columna en el archivo. Si se especifica `headings on`, el formato `ASCII` coloca el título de

	exportación y los encabezados de cada columna de métricas escrita *antes* de los primeros registros de datos.
`separator`	Especifica el carácter que se imprime entre cada campo en datos que tienen un formato `ASCII`. El carácter separador predeterminado es una barra vertical. Muchos programas prefieren una coma como separador de campos. Se puede especificar un separador como cualquier carácter imprimible o no imprimible.
`summary`	Especifica el número de minutos de datos que se van a resumen para cada intervalo de resumen. Los intervalos predeterminados son 5, 15, 30, 60, 180, 360, 720, 1440 y 10080 minutos. Si menciona otros intervalos de resumen, el programa de extracción se resume en el intervalo de resumen más próximo.
`output`	Especifica el nombre del archivo de salida en el que se escribirán los datos exportados. Es posible especificar output para cada clase o data type exportado colocando el nombre de archivo de salida 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 output.
`data type`	Especifica uno de los tipos de datos exportables: `global, application, process, disk, transaction, lvolume, netif, configuration,` 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.
`métrica`	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. *Sólo* se hará referencia a un tipo de datos si se elige exportar dicho tipo.

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`.

Salida de datos exportados

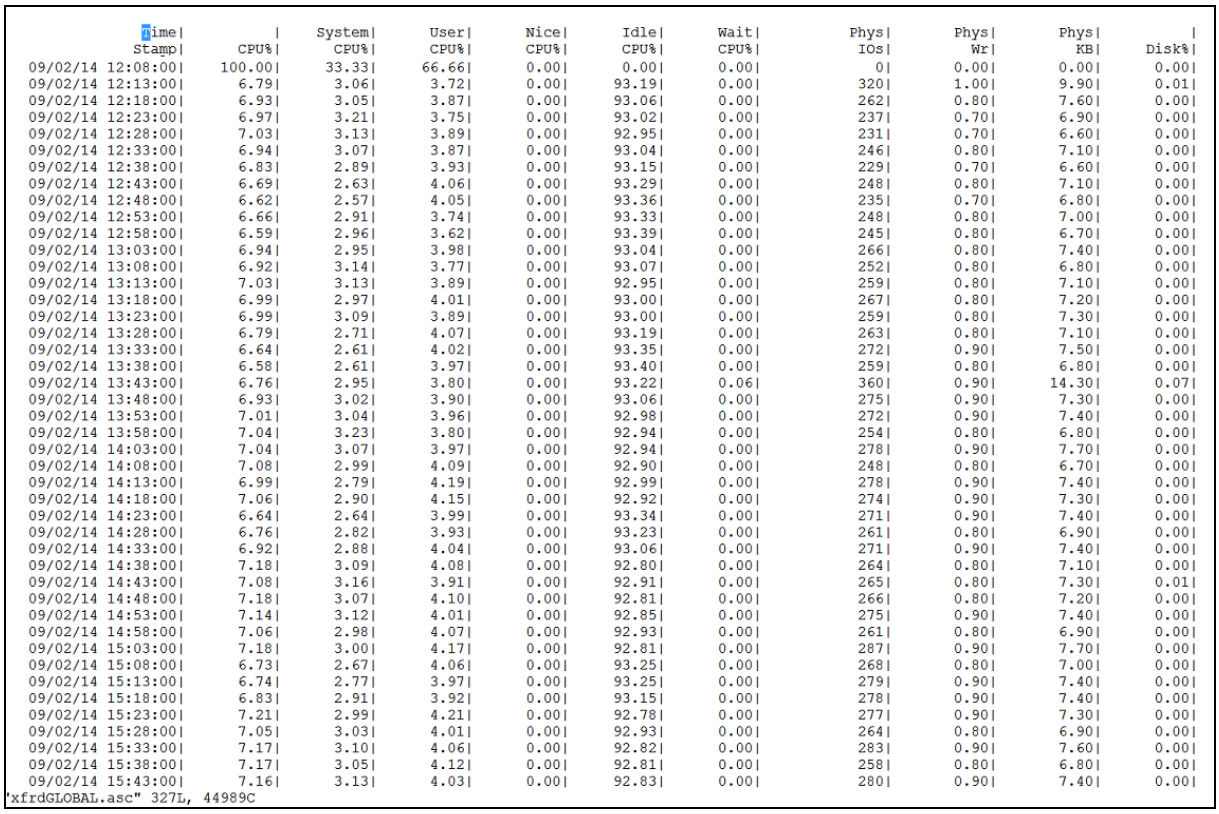
El contenido de cada archivo exportado es el siguiente:

Nombres (`application`, `netif`, `lvolume` o `transaction`)	Si se ha especificado `headings on`.
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.

Por ejemplo:

Si exporta datos de clase global, verá la salida siguiente:



Time Stamp	CPU%	System CPU%	User CPU%	Nice CPU%	Idle CPU%	Wait CPU%	Phys IOs	Phys Wr	Phys KB	Disk%
09/02/14 12:08:00	100.00	33.33	66.66	0.00	0.00	0.00	0	0.00	0.00	0.00
09/02/14 12:13:00	6.79	3.06	3.72	0.00	93.19	0.00	320	1.00	9.90	0.01
09/02/14 12:18:00	6.93	3.05	3.87	0.00	93.06	0.00	262	0.80	7.60	0.00
09/02/14 12:23:00	6.97	3.21	3.75	0.00	93.02	0.00	237	0.70	6.90	0.00
09/02/14 12:28:00	7.03	3.13	3.89	0.00	92.95	0.00	231	0.70	6.60	0.00
09/02/14 12:33:00	6.94	3.07	3.87	0.00	93.04	0.00	246	0.80	7.10	0.00
09/02/14 12:38:00	6.83	2.89	3.93	0.00	93.15	0.00	229	0.70	6.60	0.00
09/02/14 12:43:00	6.69	2.63	4.06	0.00	93.29	0.00	248	0.80	7.10	0.00
09/02/14 12:48:00	6.62	2.57	4.05	0.00	93.36	0.00	235	0.70	6.80	0.00
09/02/14 12:53:00	6.66	2.91	3.74	0.00	93.33	0.00	248	0.80	7.00	0.00
09/02/14 12:58:00	6.59	2.96	3.62	0.00	93.39	0.00	245	0.80	6.70	0.00
09/02/14 13:03:00	6.94	2.95	3.98	0.00	93.04	0.00	266	0.80	7.40	0.00
09/02/14 13:08:00	6.92	3.14	3.77	0.00	93.07	0.00	252	0.80	6.80	0.00
09/02/14 13:13:00	7.03	3.13	3.89	0.00	92.95	0.00	259	0.80	7.10	0.00
09/02/14 13:18:00	6.99	2.97	4.01	0.00	93.00	0.00	267	0.80	7.20	0.00
09/02/14 13:23:00	6.99	3.09	3.89	0.00	93.00	0.00	259	0.80	7.30	0.00
09/02/14 13:28:00	6.79	2.71	4.07	0.00	93.19	0.00	263	0.80	7.10	0.00
09/02/14 13:33:00	6.64	2.61	4.02	0.00	93.35	0.00	272	0.90	7.50	0.00
09/02/14 13:38:00	6.58	2.61	3.97	0.00	93.40	0.00	259	0.80	6.80	0.00
09/02/14 13:43:00	6.76	2.95	3.80	0.00	93.22	0.06	360	0.90	14.30	0.07
09/02/14 13:48:00	6.93	3.02	3.90	0.00	93.06	0.00	275	0.90	7.30	0.00
09/02/14 13:53:00	7.01	3.04	3.96	0.00	92.98	0.00	272	0.90	7.40	0.00
09/02/14 13:58:00	7.04	3.23	3.80	0.00	92.94	0.00	254	0.80	6.80	0.00
09/02/14 14:03:00	7.04	3.07	3.97	0.00	92.94	0.00	278	0.90	7.70	0.00
09/02/14 14:08:00	7.08	2.99	4.09	0.00	92.90	0.00	248	0.80	6.70	0.00
09/02/14 14:13:00	6.99	2.79	4.19	0.00	92.99	0.00	278	0.90	7.40	0.00
09/02/14 14:18:00	7.06	2.90	4.15	0.00	92.92	0.00	274	0.90	7.30	0.00
09/02/14 14:23:00	6.64	2.64	3.99	0.00	93.34	0.00	271	0.90	7.40	0.00
09/02/14 14:28:00	6.76	2.82	3.93	0.00	93.23	0.00	261	0.80	6.90	0.00
09/02/14 14:33:00	6.92	2.88	4.04	0.00	93.06	0.00	271	0.90	7.40	0.00
09/02/14 14:38:00	7.18	3.09	4.08	0.00	92.80	0.00	264	0.80	7.10	0.00
09/02/14 14:43:00	7.08	3.16	3.91	0.00	92.91	0.00	265	0.80	7.30	0.01
09/02/14 14:48:00	7.18	3.07	4.10	0.00	92.81	0.00	266	0.80	7.20	0.00
09/02/14 14:53:00	7.14	3.12	4.01	0.00	92.85	0.00	275	0.90	7.40	0.00
09/02/14 14:58:00	7.06	2.98	4.07	0.00	92.93	0.00	261	0.80	6.90	0.00
09/02/14 15:03:00	7.18	3.00	4.17	0.00	92.81	0.00	287	0.90	7.70	0.00
09/02/14 15:08:00	6.73	2.67	4.06	0.00	93.25	0.00	268	0.80	7.00	0.00
09/02/14 15:13:00	6.74	2.77	3.97	0.00	93.25	0.00	279	0.90	7.40	0.00
09/02/14 15:18:00	6.83	2.91	3.92	0.00	93.15	0.00	278	0.90	7.40	0.00
09/02/14 15:23:00	7.21	2.99	4.21	0.00	92.78	0.00	277	0.90	7.30	0.00
09/02/14 15:28:00	7.05	3.03	4.01	0.00	92.93	0.00	264	0.80	6.90	0.00
09/02/14 15:33:00	7.17	3.10	4.06	0.00	92.82	0.00	283	0.90	7.60	0.00
09/02/14 15:38:00	7.17	3.05	4.12	0.00	92.81	0.00	258	0.80	6.80	0.00
09/02/14 15:43:00	7.16	3.13	4.03	0.00	92.83	0.00	280	0.90	7.40	0.00

"xfrdGLOBAL.asc" 327L, 44989C

Nota:

Aunque las métricas **gbl_statdate** y **gbl_stattime** no se seleccionan en reptail o reptall, los archivos exportados siempre contienen la marca de fecha y hora en el formato de 24 horas.

Con HPE Operations Agent 12.01, los encabezados de métrica mostrados en la salida de Extract son coherentes con los encabezados de métricas mostrados con las herramientas en tiempo real, como perfd y Glance.

Producción de un archivo de salida personalizado

Se proporciona el archivo de plantilla de exportación `reptall` con Componente Performance Collection. Este archivo se encuentra en el directorio `/var/opt/perf/`. Puede usar `reptall` para personalizar el archivo de salida, como se menciona después:

Personalizar archivo de exportación:

Para personalizar el archivo de exportación, siga estos pasos:

1. En el archivo de plantilla de exportación, quite las marcas de comentario de las métricas que desee exportar.

2. Guarde y exporte el archivo.

Notas sobre el formato ASCII

El formato ASCII (o texto) es el más adecuado para copiar archivos en una impresora o terminal. 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`. 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`.

Nota: 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`.

Sugerencia: 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.

Comandos del programa extract

En el presente capítulo se describen los comandos del programa **Extract** con ejemplos.

Tabla 6: Comandos del programa extract

Opciones de comando	Descripción
application	Use la opción `application` para especificar el tipo de datos de aplicación que se exportarán. **Por ejemplo:** Para exportar los datos detallados de la aplicación, ejecute el siguiente comando: `extract -a -r /var/opt/perf/myrept -xp` Como no se ha especificado ningún archivo de salida, los datos de aplicación se exportan a `xfrdAPPLICATION.asc`. El archivo de salida contiene todas las métricas de la aplicación especificadas en el archivo de plantilla de exportación myrept.
cpu	Use la opción `cpu` para especificar el nivel de resumen de CPU. **Por ejemplo:** Para exportar los datos detallados de la CPU que se recopilaron desde el 26 de julio de 2014, ejecute el siguiente comando: `extract -u -b 7/26/14 -xp` En este ejemplo, como no se ha especificado ningún archivo de salida, los datos de CPU se exportan a `xfrdCPU.asc`. Puesto que no se especifica ningún archivo de plantilla de exportación, se usa el predeterminado: *reptfile*. Todas las métricas de CPU especificadas en *reptfile* se incluyen en el archivo de salida.
disk	Use la opción `disk` para especificar el tipo de datos del dispositivo de disco que se exportarán. **Por ejemplo:** Para exportar los datos detallados del disco que se recopilaron desde el 05 de julio de 2014, ejecute el siguiente comando:

Tabla 6: Comandos del programa extract, continuación

	`extract -d -b 7/5/14 -xp` En este ejemplo, como no se ha especificado ningún archivo de salida, los datos detallados del disco se exportan a `xfrdDISK.asc`. Puesto que no se especifica ningún archivo de plantilla de exportación, se usa el predeterminado: *reptfile*. Todas las métricas de disco especificadas en *reptfile* se incluyen en el archivo de salida-
export	Use la opción `export` para iniciar el proceso de copiar datos en un archivo de salida. **Por ejemplo:** Para exportar los datos detallados globales, ejecute el siguiente comando: `extract -xp -g` La clase global de datos se exporta al archivo `xfrdGLOBAL.asc`.
filesystem	Use la opción `filesystem` para especificar el nivel de resumen de los datos del sistema de archivo que se van a exportar. **Por ejemplo:** Para exportar los datos detallados del sistema de archivo que se recopilaron desde el 26 de julio de 2014, ejecute el siguiente comando: `extract -y -b 7/26/14 -xp` Los datos del sistema de archivo se exportaron en `xfrdFILESYSTEM.asc`. Puesto que no se especifica ningún archivo de plantilla de exportación, se usa el predeterminado: reptfile. Todas las métricas del sistema de archivo especificadas en `reptfile` se incluyen en el archivo de salida.
global	Use la opción `global` para especificar la cantidad de datos globales que se exportarán.

Tabla 6: Comandos del programa extract, continuación

	Por ejemplo: Para exportar los datos detallados globales, ejecute el siguiente comando: `extract -g -r /var/opt/perf/myrept -f myout -xp` Los datos globales se exportan al archivo de salida denominado `myout`. El archivo de salida contiene todas las métricas globales especificadas en el archivo de plantilla de exportación myrept.
help	Use la opción `help` para acceder a la ayuda en línea.
lvolume	Use la opción `lvolume` para especificar el tipo de datos del volumen lógico que se exportarán. **Nota:** Esta opción solo se admite en los sistemas HP-UX y Linux. **Por ejemplo:** Para exportar los datos detallados del volumen lógico, ejecute el siguiente comando: `extract -z -r /var/opt/perf/myrept -xp` Como no se ha especificado ningún archivo de salida, los datos del volumen lógico se exportan a `xfrdVOLUME.asc`. El archivo de salida contiene las métricas especificadas en el archivo de plantilla de exportación myrept.
netif	Use la opción `netif` para especificar el tipo de datos de la interfaz de red lógica (LAN) que se exportarán. **Por ejemplo:** Para exportar los datos detallados de la interfaz de red, ejecute el siguiente comando: `extract -n -r /var/opt/perf/myrept -xp` Como no se ha especificado ningún archivo de salida, los datos de la interfaz de red se exportan a `xfrdNETIF.asc`. El archivo de salida contiene las métricas especificadas en el

Tabla 6: Comandos del programa extract, continuación

	archivo de plantilla de exportación myrept.
process	Use la opción `process` para especificar si se exportarán o no los datos del proceso. **Por ejemplo:** Para exportar los datos detallados del proceso, ejecute el siguiente comando: `extract -p -r /var/opt/perf/myrept -xp` Como no se ha especificado ningún archivo de salida, los datos del proceso se exportan a `xfrdPROCESS.asc`. El archivo de salida contiene todas las métricas del proceso especificadas en el archivo de plantilla de exportación myrept.

start	Use la opción `start` para establecer una fecha y hora de inicio para las funciones de exportación. 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.
	Las opciones `First` y `last` se admiten cuando se usan con los comandos de inicio (`-b`) o de detención (`-e`).
	Por ejemplo:
	1. Para exportar los datos detallados globales desde el 5 de junio de 1999, a las 8:00 a.m., ejecute el siguiente comando:
	`extract -g -b 06/05/99 8:00 -f myout -xp`
	Los datos globales se exportan al archivo de salida denominado `myout.`
	Puesto que no se especifica ningún archivo de plantilla de exportación, se usa el predeterminado: reptfile. Todas las métricas globales especificadas en reptfile se incluyen en el archivo de salida.
	2. Para exportar solo los últimos datos globales, ejecute el siguiente comando:
	`extract -g -b last-f myout -xp`
	3. Para exportar los primeros y últimos datos globales al

Tabla 6: Comandos del programa extract, continuación

	archivo `xfrdGLOBAL.asc`, ejecute el siguiente comando: `extract -g -b first -e last -xp`
stop	Use la opción `stop` para finalizar una función de exportación 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. **Por ejemplo:** Para exportar los datos detallados globales desde el 5 de junio de 2014, a las 8:00 a.m., al 5 de junio de 2014 a las 5:00 p.m., ejecute el siguiente comando: `extract -g -b 6/5/14 8:00 -e 6/5/14 17:00  -f myout -xp` Los datos globales se exportan al archivo de salida denominado `myout.` Puesto que no se especifica ningún archivo de plantilla de exportación, se usa el predeterminado: **reptfile**. Todas las métricas globales especificadas en reptfile se incluyen en el archivo de salida.
transaction	Use la opción `transaction` para especificar el tipo de datos de transacción que se exportará. **Por ejemplo:** Para exportar los datos detallados de la transacción, ejecute el siguiente comando: `extract -t -r /var/opt/perf/myrept -xp` Como no se ha especificado ningún archivo de salida, los datos de la transacción se exportan a `xfrdTRANSACTION.asc`. El archivo de salida contiene todas las métricas de la transacción especificadas en el archivo de plantilla de exportación myrept.
host bus adapter	Use la opción `host bus adapter` para especificar la cantidad de datos HBA que se exportarán. **Por ejemplo:**

Tabla 6: Comandos del programa extract, continuación

	Para exportar los datos detallados de HBA que se recopilaron desde el 26 de julio de 2014, ejecute el siguiente comando: `extract -h -b 7/26/14 -xp` En este ejemplo, como no se ha especificado ningún archivo de salida, los datos HBA se exportan a `xfrdHBA.asc`. **Nota:** • El archivo de plantilla de exportación predeterminado, reptfile, no contiene las métricas HBA. Para exportar los datos HBA, agregue manualmente las métricas HBA. • Esta opción solo se admite en los sistemas Linux, Windows y HP-UX.
classname y logfile	Use el comando `classname` para especificar la clase de datos DSI que se exportarán. **Sintaxis** extract -xp -C <nombre_clase_DSI> -l <ruta_archivo_registro>
core	Use el comando `core` para especificar el tipo de datos bycore que se exportarán. **Nota:** Esta opción se admite en los sistemas Linux, Windows, HP-UX y Solaris. **Por ejemplo:** Para exportar los datos detallados de la aplicación, ejecute el siguiente comando: `extract -x -r /var/opt/perf/myrept -xp` El archivo de salida contiene todas las métricas de la aplicación especificadas en el archivo de plantilla de exportación myrept.

Capítulo 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 proporciona 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 HPE 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` *<nombre del sistema>*

   donde *<nombre del sistema>* es el nombre de dominio completo del sistema remoto.

   o

   `cpsh -n` *<dirección IP>*

   donde *<dirección IP>* es la dirección IP del sistema remoto.

   Nota: Al abrir el símbolo del sistema `cpsh` de un sistema remoto, asegúrese de que el proceso `perfd` se ejecute en el sistema remoto. Puede impedir que otros sistemas accedan a los datos de rendimiento del sistema local a través de la utilidad `cpsh`. Para obtener más información, consulte "Restricción de acceso" en la página 64.

   Se abrirá el símbolo del sistema `cpsh`.

Para ver datos de métricas en un formato bien estructurado, ejecute el comando `cpsh` con la opción `-t`.

Por ejemplo:

```
cpsh -t
```

o

`cpsh -n` *<nombre del sistema>* `-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. At the `cpsh` prompt, type **class** *<clase de métrica>*.
2. Para obtener una lista de todas las métricas actualmente establecidas 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. At the cpsh prompt, type **class** *<clase de métrica>*.
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:

   Tras mostrarse el símbolo del sistema cpsh, escriba **delete** *<nombre_de_métrica>*.

   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:

   Tras mostrarse el símbolo del sistema cpsh, escriba **add** *<nombre_de_métrica>*.

   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 cpsh.
2. At the cpsh prompt, type **class** *<clase de métrica>*.
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 cpsh.
2. At the cpsh prompt, type **class** *<clase de métrica>*.
3. Escriba **init** *<nombre_de_métrica> <nombre_de_métrica> <nombre_de_métrica> ....*

   La clase de métrica especificada sólo incorporará la métrica especificada con el comando init.

Ver Ayuda de métrica

El símbolo de sistema cpsh 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 cpsh.
2. Escriba class *<clase de métrica>*.

3. Escriba **help** <*nombre_de_métrica*> en el indicador `cpsh`. Se mostrará una descripción de detalles de la métrica.

Ver datos de métrica resumidos

El símbolo del sistema cpsh 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 `cpsh`.
2. Tras mostrarse el símbolo del sistema `cpsh`, escriba **class gbl** o **class tbl**.
3. Escriba **summ** <*intervalo*>. En este caso, <*intervalo*> es el intervalo de resumen especificado en segundos. <*intervalo*> debe ser un intervalo de recopilación múltiple del servidor **perfd** al que está conectado `cpsh`.

   La utilidad `cpsh` muestra las siguientes medidas de los valores de métrica que pertenecen a la clase de métrica seleccionada:

   - Máximo
   - Mínimo
   - Promedio
   - Desviación estándar

Habilitación de las condiciones de umbral y de filtro

Puede establecer las opciones de umbral y de filtro en función de sus requisitos en el proceso Perfd y ver los datos disponibles con los criterios de calificación establecidos.

Establezca las opciones siguientes:

Umbral - Esta opción solo está disponible para las métricas de la clase GLOBAL. Puede especificar el umbral global y los datos disponibles sobre el valor de umbral global establecido.

Filtro - La opción de filtro está disponible para las métricas de todas las clases.

Puede establecer las condiciones como:

- **filtro** <`métrica`> <`operador`> <`valor`>
- **umbral** <`métrica`> <`operador`> <`valor`>

También puede establecer las dos condiciones para ver solo los datos que cumplen el criterio establecido.

Ejemplo 1: Listar el número de procesos cuya utilización es mayor del 4 por ciento.

Use la condición de filtro para ver los datos.

Para ver los datos filtrados, siga estos pasos:

1. Abra el símbolo del sistema cpsh.
2. En el indicador cpsh, escriba **class proc**.
3. Escriba **filter proc_cpu_total_util > 4**.

   La utilidad cpsh muestra los procesos cuya utilización de la CPU es superior al 4 por ciento.
4. Escriba **push**.

El resultado aparece como:

```
User   App     Interest        Process         ProcessName

Name   PID     ID Reason       Start Time      CPU Name

root  1028     4       05/07/2013 07:17:35     5.2 find
```

Ejemplo 2: Listar el número de procesos cuya utilización es mayor del 4 por ciento y el umbral está establecida en 1 por ciento.

Establezca la opción de umbral y después la opción de filtro para ver los datos.

Para ver los datos filtrados, siga estos pasos:

1. Abra el símbolo del sistema cpsh.
2. En el indicador cpsh, escriba **class proc**.
3. Escriba **threshold proc_cpu_total_util > 1**.
4. Escriba **filter proc_cpu_total_util > 3**.

   La utilidad cpsh muestra los procesos cuyo valor de umbral está establecido en un valor superior al 1 por ciento y también cuya utilización de la CPU es mayor del 3 por ciento.
5. Escriba **push**.

El resultado aparece como:

```
User   App     Interest        Process         ProcessName

Name   PID     ID Reason       Start Time      CPU Name

root  1028     4       05/07/2013 07:17:35     4.9 find
```

Capítulo 9: Creación de recopilaciones de contadores de rendimiento en Windows

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. Extended Collection Builder and Manager (ECBM) se utiliza para seleccionar contadores de rendimiento específicos para crear recopilaciones de datos.

Nota: Asegúrese de que el proceso **oacore** se está ejecutando antes de registrar una directiva e iniciar una recopilación.

Creación de una recopilación de contadores de rendimiento

IC para crear una recopilación, elija **ECB-ECM (Extended Collection Builder and Manager)** en el menú **Inicio** o ejecute el comando `mwecbm` con el símbolo del sistema para abrir **Extended Collection Builder and Manager**.

Aparece la ventana Extended Collection Builder and Manager con una lista de objetos de Windows en el panel izquierdo. Para obtener instrucciones acerca de la creación de las recopilaciones, elija **Temas de ayuda** en el menú Ayuda en la ventana Extended Collection Builder and Manager.

Después de crear sus recopilaciones de los contadores de rendimiento de Windows, use el panel **Administrador de recopilaciones ampliadas** para registrar, iniciar y detener recopilaciones nuevas y existentes. También puede usar la opción **Administrar** para iniciarlas, detenerlas o eliminarlas.

Administración de una recopilación de contadores de rendimiento

Para administrar las recopilaciones de datos, utilice el panel **Administrador de recopilaciones ampliadas**, en la parte inferior de Extended Collection Builder and Manager. Inicialmente, no aparece ninguna recopilación porque debe registrar una antes de poder empezar a recopilar datos.

Después de registrar o almacenar la recopilación que ha creado, en el panel Administrador de recopilaciones ampliadas se muestra una lista de las recopilaciones actuales. En el panel Administrador de recopilaciones ampliadas también se muestra el estado de cada recopilación y se puede ver información (propiedades) acerca de la recopilación. Para obtener instrucciones acerca de la administración de las recopilaciones, elija **Temas de ayuda** en el menú **Ayuda** en la ventana Extended Collection Builder and Manager.

Sugerencias para el uso de Extended Collection Builder and Manager

El archivo `<Dir_instalación>\paperdocs\ovpa\C\monxref.txt` contiene una referencia cruzada de las métricas de Componente Performance Collection para los comandos y contadores de rendimiento de Windows. El registro de datos mediante Extended Collection Builder and Manager para las métricas ya recopiladas por Componente Performance Collection genera una sobrecarga del sistema adicional.

Al utilizar Extended Collection Builder para crear recopilaciones, los nombres de métricas predeterminados se asignan a los contadores de rendimiento de Windows para su uso interno con Componente Performance Collection. Normalmente, estos nombres predeterminados no son significativos ni fáciles de descifrar. Para que los nombres de las métricas tengan más sentido o coincidan con los nombres de las métricas suministrados por su aplicación de origen, modifique sus atributos haciendo clic con el botón derecho o doble clic en la métrica después de arrastrarla del panel izquierdo al derecho en la ventana de Extended Collection Builder and Manager. (Consulte la ayuda en línea de Extended Collection Builder and Manager para ver instrucciones detalladas).

Si inicia 61 o más recopilaciones, las recopilaciones por encima de la 60 tendrán un estado de error. Esto puede causar problemas con otras recopilaciones.

Si recopila métricas del disco lógico de un sistema configurado con **Wolfpack**, debe reiniciar la recopilación para recopilar datos de cualquier nueva instancia de disco no presente cuando se registra la recopilación.

La eliminación correcta de las recopilaciones requiere reiniciar Componente Performance Collection después de eliminar la recopilación. Si no se ha reiniciado Componente Performance Collection, puede obtener un error durante la operación de eliminación. Normalmente, este error significa que algunos archivos no se han eliminado correctamente. Puede ser necesario eliminar manualmente cualquier archivo y directorio que permanezca después de reiniciar Componente Performance Collection.

Extended Collection Builder and Manager puede informar acerca de valores que faltan para algunas métricas con contadores de caché. El problema puede producirse en determinadas circunstancias cuando un valor de métrica genera un desbordamiento. También se envía un mensaje al archivo de estado de ECBM. Puede resolver el problema reiniciando la recopilación.

Nota: ECBM no puede procesar un valor de métrica por encima de 2147483647.

Hay explicaciones de conceptos de Extended Collection Builder and Manager e instrucciones sobre la creación y visualización de recopilaciones de datos en la ayuda en línea de Extended Collection Builder and Manager. Para ver la ayuda en línea, en el escritorio seleccione **Inicio --> Todos los programas --> HPE Operations Agent --> Performance Collection Component --> ECB-ECM Online Help**. Puede seleccionar **Recopilaciones ampliadas** en el menú Agente de la ventana principal de Componente Performance Collection y elegir **Temas de ayuda** en el menú Ayuda de la ventana de Extended Collection Builder and Manager. Puede usar la ayuda en línea seleccionando el botón **Ayuda** en los cuadros de diálogo que aparecen en Extended Collection Builder and Manager.

Administración de ECBM desde la línea de comandos

Puede ejecutar el programa ECBM del directorio <rpmtools>\bin con el símbolo del sistema de Windows.

- Las recopilaciones se pueden administrar desde la línea de comandos con el siguiente comando:

```
\<InstallDir>\bin\mwcmcmd.exe
```

- Para mostrar las diversas opciones, escriba el siguiente comando:

```
\<InstallDir>\bin\mwcmcmd /?
```

- Para iniciar las recopilaciones detenidas, escriba el siguiente comando:

mwcmcmd start <nombres_recopilaciones>

- Para iniciar una nueva recopilación desde una directiva de instancias de variables, escriba el siguiente comando:

mwcmcmd start <nombre_directiva> <nombre_recopilación> <instancias> [options]

Están disponibles las siguientes opciones:

-i <intervalo_muestreo>: cambia el intervalo de muestreo (segundos)

```
-l <nombre_ruta_archivo_registro>: cambia la ubicación del registro
predeterminado

-a <archivo_alarma>: cambia el archivo de definiciones de alarma
```

- Para detener las recopilaciones activas, escriba el siguiente comando:

```
mwcmcmd stop <nombres_recopilaciones>
```

- Para registrar un archivo de directivas, escriba el siguiente comando:

mwcmcmd register <archivo_directiva> <nombre_recopilación/directiva> [options]

Las siguientes opciones solo están disponibles al registrar un archivo de directiva de estancia fija:

```
-i <intervalo_muestreo>: cambia el intervalo de muestreo (segundos)

-l <nombre_ruta_archivo_registro>: cambia la ubicación del registro
predeterminado

-a <archivo_alarma>: cambia el archivo de definiciones de alarma
```

- Para eliminar una sola recopilación:

mwcmcmd delete <nombre_recopilación/directiva> [option]

La siguiente opción está disponible al eliminar una recopilación:

-r : reinicia el componente HPE Performance Collection

Nota: Al eliminar una directiva o recopilación, se eliminarán todos los archivos de base de datos asociados. Además, el proceso **oacore** se detiene mientras se elimina una recopilación, y se inicia automáticamente cuando se elimina.

- Para eliminar múltiples recopilaciones o directivas:

mwcmcmd delete { <nombres_recopilación/directiva> | -c | -all }

```
-c : elimina TODAS las recopilaciones

-a : elimina TODAS las recopilaciones y directivas
```

Nota: Al eliminar más de una directiva o recopilación al mismo tiempo, el componente Performance Collection se reiniciará automáticamente y se eliminarán todos los archivos de base de datos asociados.

- Para enumerar todas las recopilaciones y directivas registradas, escriba el siguiente comando:

```
mwcmcmd list
```

- Para enumerar las propiedades de una recopilación o directiva, escriba el siguiente comando:

```
mwcmcmd properties <nombre_recopilación/directiva>
```

- Para enumerar los objetos de instancias de variables en una directiva, escriba:

mwcmcmd objects <nombre_directiva>

Capítulo 10: Información general del proceso de líneas base

El proceso de líneas base se utiliza para calcular y proporcionar valores de referencia basados en datos históricos.[1] almacenados en el almacén de datos de métricas. Para calcular los datos de línea base durante un período de tiempo específico, se usan los datos de métrica recopilados en los períodos de tiempo correspondientes de las semanas anteriores. Los datos de línea base incluyen los valores mínimo, máximo, media y desviación estándar. Los datos de línea base se calculan al final de cada hora y se almacenan en el almacén de datos de métricas.

Los datos de la línea de base se usan para:

- Proporcionar valores de referencia para monitorizar el rendimiento diario
- Proporcionar valores de referencia para analizar tendencias de rendimiento
- Establecer dinámicamente los valores de umbral óptimos para analizar el patrón de la utilización de recursos

Los datos de línea base calculados por HPE Operations Agent se usan por la directiva SI-AdaptiveThresholdingMonitor para monitorizar el rendimiento y el uso de recursos. Para obtener más información, consulte el tema *Adaptive Thresholding Policies* en la guía *HPE Operations Smart Plug-in for System Infrastructure User Guide.*

Configuración de la línea base en el nodo HPE Operations Agent

El proceso de líneas base no está habilitado de manera predeterminada en el nodo de HPE Operations Agent. Para configurar un proceso de líneas base en un nodo de HPE Operations Agent, siga estos pasos:

1. Actualice el archivo `baseline.cfg`.

   Nota: `baseline.cfg` es un archivo de texto plano usado para definir las métricas que desea monitorizar.

   a. Inicie sesión en el nodo de HPE Operations Agent.
   b. Vaya al directorio siguiente:

      En Windows:

      `%ovdatadir%`

En HP-UX/Linux/Solaris:

`/var/opt/perf/`

c. Actualice el archivo `baseline.cfg` y defina las métricas que desee monitorizar en el formato siguiente:

`<Class>:<Metric>`

En este ejemplo:

- `<Class>` es la clase de la métrica.
- `<Metric>` es la métrica para la que se deben calcular los datos de línea base.

Nota: Los datos de línea base solo se calculan para las métricas gauge.

Por ejemplo:

Global:GBL_CPU_TOTAL_UTIL

En este ejemplo:

`<clase>`	Global
`<Métrica>`	GBL_CPU_TOTAL_UTIL

2. Habilitar el proceso de línea base en el nodo de HPE Operations Agent
   a. Vaya al directorio siguiente:

   En Windows x64:

   `%ovinstalldir%bin\win64`

   En Windows x86:

   `%ovinstalldir%bin`

   En AIX:

   `/usr/lpp/OV/bin`

   En HP-UX/Linux/Solaris:

   `/opt/OV/bin`

   b. Ejecute el comando siguiente:

   `ovconfchg –ns oacore –set ENABLE_BASELINE TRUE`

   De manera predeterminada, el valor es FALSE.

Los datos de línea base se calculan solo al final de cada hora. Si desea que los datos de línea base se calculen inmediatamente después de habilitar la línea

base, debe reiniciar el proceso **oacore**. Ejecute el comando siguiente para reiniciar **oacore**:

```
ovc -restart oacore
```

Nota: También puede usar las configuraciones de XPL para habilitar la línea base. Siga estos pasos:

1. En la consola de HPOM seleccione **Gestión de directivas → Grupos de directivas → Gestión de infraestructura → v12.0 → Configuración y umbrales → Configuración de agente → OPC_PERL_INCLUDE_INSTR_DIR**
2. Establezca la variable ENABLE_BASELINE en TRUE y despliegue la directiva en todos los nodos deseados.

Para obtener más información, consulte la guía *HPE Operations Smart Plug-in for System Infrastructure User Guide.*

Monitorización de métricas

Nota: Una vez calculados los datos de la línea base, utilice la directiva de umbral de medición para monitorizar las métricas deseadas. Puede usar las directivas SPI de infraestructura predeterminadas, como la directiva SI-ConfigureBaselining y SI-AdaptivethresholdingMonitor para monitorizar las métricas de líneas base. Para obtener más información, consulte la guía *HPE Operations Smart Plug-in for System Infrastructure User Guide.*

En función de las clases definidas en el archivo `baseline.cfg`, se crean las clases correspondientes de la línea base en el Almacén de datos de métricas.

Nota: Las clases de línea base finalizan con _BASELINE.

Por ejemplo:

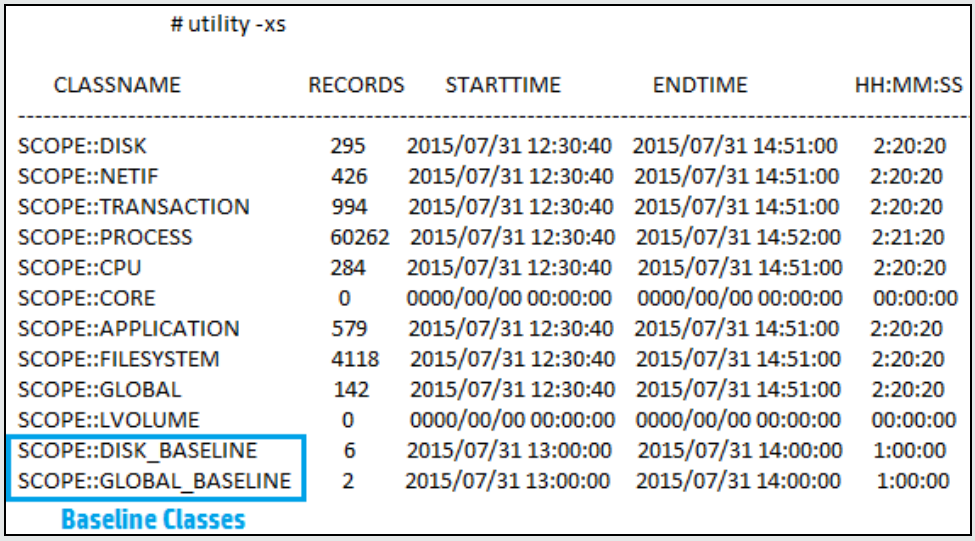
Si especifica las métricas siguientes en el archivo `baseline.cfg`:

```
Disk:BYDSK_UTIL

Global:GBL_CPU_TOTAL_UTIL
```

Se crean dos clases de líneas base:

```
DISK_BASELINE

GLOBAL_BASELINE
```

Ejecute el comando `utility -xs` para ver la información sobre las clases de líneas base:



utility -xs

CLASSNAME	RECORDS	STARTTIME	ENDTIME	HH:MM:SS
SCOPE::DISK	295	2015/07/31 12:30:40	2015/07/31 14:51:00	2:20:20
SCOPE::NETIF	426	2015/07/31 12:30:40	2015/07/31 14:51:00	2:20:20
SCOPE::TRANSACTION	994	2015/07/31 12:30:40	2015/07/31 14:51:00	2:20:20
SCOPE::PROCESS	60262	2015/07/31 12:30:40	2015/07/31 14:52:00	2:21:20
SCOPE::CPU	284	2015/07/31 12:30:40	2015/07/31 14:51:00	2:20:20
SCOPE::CORE	0	0000/00/00 00:00:00	0000/00/00 00:00:00	00:00:00
SCOPE::APPLICATION	579	2015/07/31 12:30:40	2015/07/31 14:51:00	2:20:20
SCOPE::FILESYSTEM	4118	2015/07/31 12:30:40	2015/07/31 14:51:00	2:20:20
SCOPE::GLOBAL	142	2015/07/31 12:30:40	2015/07/31 14:51:00	2:20:20
SCOPE::LVOLUME	0	0000/00/00 00:00:00	0000/00/00 00:00:00	00:00:00
SCOPE::DISK_BASELINE	6	2015/07/31 13:00:00	2015/07/31 14:00:00	1:00:00
SCOPE::GLOBAL_BASELINE	2	2015/07/31 13:00:00	2015/07/31 14:00:00	1:00:00

Baseline Classes

Para cada métrica especificada en el archivo `baseline.cfg`, se crean dieciséis métricas de líneas base.

Por ejemplo:

Si especifica GBL_CPU_TOTAL_UTIL in en el archivo `baseline.cfg`, se crean dieciséis métricas de líneas base globales.

Nota: Puede ejecutar el comando `ovcodautil-obj` para ver las métricas de líneas base.

Ejecute el comando siguiente para ver las métricas de líneas base creadas para la clase GLOBAL_BASELINE:

```
ovcodautil -ds SCOPE -o GLOBAL_BASELINE -obj
```

```
GLOBAL_BASELINE        ATT   I64   INSTANCEID
GLOBAL_BASELINE        KEY   UTF8  GBL_SYSTEM_ID
GLOBAL_BASELINE        GGE   R64   GBL_CPU_TOTAL_UTIL_MAX_BL
GLOBAL_BASELINE        GGE   R64   GBL_CPU_TOTAL_UTIL_MIN_BL
GLOBAL_BASELINE        GGE   R64   GBL_CPU_TOTAL_UTIL_AVG_BL
GLOBAL_BASELINE        GGE   R64   GBL_CPU_TOTAL_UTIL_STDDEV_BL
GLOBAL_BASELINE        GGE   R64   GBL_CPU_TOTAL_UTIL_STDDEV_BL_LASTHOUR
GLOBAL_BASELINE        GGE   I64   GBL_CPU_TOTAL_UTIL_BL_SAMPLESIZE
GLOBAL_BASELINE        GGE   R64   GBL_CPU_TOTAL_UTIL_MAX_BL_LASTHOUR
GLOBAL_BASELINE        GGE   R64   GBL_CPU_TOTAL_UTIL_MIN_BL_LASTHOUR
GLOBAL_BASELINE        GGE   R64   GBL_CPU_TOTAL_UTIL_AVG_BL_LASTHOUR
GLOBAL_BASELINE        GGE   I64   GBL_CPU_TOTAL_UTIL_BL_SAMPLESIZE_LASTHOUR
GLOBAL_BASELINE        GGE   I64   GBL_CPU_TOTAL_UTIL_S0
GLOBAL_BASELINE        GGE   I64   GBL_CPU_TOTAL_UTIL_S0_LASTHOUR
GLOBAL_BASELINE        GGE   R64   GBL_CPU_TOTAL_UTIL_S1
GLOBAL_BASELINE        GGE   R64   GBL_CPU_TOTAL_UTIL_S1_LASTHOUR
GLOBAL_BASELINE        GGE   R64   GBL_CPU_TOTAL_UTIL_S2
GLOBAL_BASELINE        GGE   R64   GBL_CPU_TOTAL_UTIL_S2_LASTHOUR
```

Los tipos de resumen son media (Avg), mínimo (Min), máximo (Max), desviación estándar (STDDEV) y tamaño de muestra (SAMPLESIZE).

Las métricas de líneas base que acaban por _BL incluyen los datos históricos (junto con los datos de la última hora).

Las métricas de líneas base que acaban por _BL_LASTHOUR incluyen solo los datos de la última hora.

Puede crear una directiva de umbral de medición para comparar los datos de la línea base de última hora con los datos históricos.

Por ejemplo: Puede usar la directiva de umbral de medición para comparar GBL_CPU_TOTAL_UTIL_AVG_BL_LASTHOUR con GBL_CPU_TOTAL_UTIL_AVG_BL.

Las métricas de líneas base S0, S1 y S2 se usan para calcular la desviación estándar.

Nota: Para detener la funcionalidad de proceso de líneas base en el nodo de HPE Operations Agent, ejecute el siguiente comando:

```
ovconfchg –ns oacore –set ENABLE_BASELINE FALSE
```

¿Cómo solucionar problemas cuando el proceso de líneas base no funciona?

Siga estos pasos:

1. **Compruebe que la línea base está habilitada.**
   a. Inicie sesión en el nodo HPE Operations Agent .
   b. Vaya al directorio siguiente:
      En las versiones de Windows de 32 bits:
      `"%ovinstalldir%/bin"`
      En las versiones de Windows de 64 bits:
      `"%ovinstalldir%/bin/win64"`
      En AIX:
      `/usr/lpp/OV/bin`
      En HP-UX/Linux/Solaris:
      `/opt/OV/bin`
   c. Ejecute el comando siguiente:
      `ovconfget oacore ENABLE_BASELINE`

   Si la línea base está habilitada, la salida es **TRUE**.
2. **Compruebe en la salida de ovcodautil las métricas de línea base.**
   a. Vaya al directorio siguiente:
      En las versiones de Windows de 32 bits:
      `"%ovinstalldir%/bin"`
      En las versiones de Windows de 64 bits:
      `"%ovinstalldir%/bin/win64"`
      En AIX:
      `/usr/lpp/OV/bin`
      En HP-UX/Linux/Solaris:
      `/opt/OV/bin`
   b. Ejecute el comando siguiente:
      `ovcodautil -ds SCOPE –o <CLASS>_BASELINE –rawonly`

      Por ejemplo:

      `ovcodautil -ds SCOPE –o GLOBAL_BASELINE –rawonly`

   Después, compruebe la salida para ver si la métrica de línea base deseada aparece en la lista.
3. **Compruebe si se mencionan las métricas para las que se van a calcular los datos de la línea base en el archivo de configuración de la línea base.**

a. Compruebe si la métrica se menciona en formato correcto en los archivos siguientes:

En Windows:

`%ovdatadir%/baseline.cfg`

En HP-UX/Linux/Solaris/AIX:

`/var/opt/perf/baseline.cfg`

Nota:

Las métricas para las que se van a calcular los datos de la línea base deben tener uno de los formatos siguientes:

```
[Baseline]
<Class>:<Metric>
```

Por ejemplo:

Global:GBL_CPU_TOTAL_UTIL

4. **Compruebe si las métricas mencionadas en el archivo de configuración de la línea base son válidas en la plataforma que está usando.**

Nota: Los datos de línea base solo se calculan para las métricas gauge.

a. Vaya al directorio siguiente:

En las versiones de Windows de 32 bits:

`"%ovinstalldir%/bin"`

En las versiones de Windows de 64 bits:

`"%ovinstalldir%/bin/win64"`

En AIX:

`/usr/lpp/OV/bin`

En HP-UX/Linux/Solaris:

`/opt/OV/bin`

b. Ejecute el comando siguiente:

`ovcodautil -ds SCOPE –o <clase> -m <métricas> -rawonly`

En este ejemplo:

<*Class*> es la clase de la métrica.

<*Metrics*> es la métrica para la que se deben calcular los datos de línea base.

Compruebe si las métricas mencionadas en el archivo de configuración de la línea base se enumeran en la salida.

[1]Los datos históricos consisten en los datos recopilados hasta la hora anterior y almacenados en el almacén de datos de métricas.

Capítulo 11: Información general de la resolución de nodos

Un nodo puede tener una o varias interfaces de red. La dirección IP asociada a cada interfaz de red puede tener un nombre de host. HPE Operations Agent selecciona el nombre de host usando las API específicas del sistema operativo en la configuración del sistema (ejemplo, gethostname /getaddrinfo /getnameinfo). Configura automáticamente el parámetro OPC_NODENAME que proporciona el valor del nombre de host local usado por HPE Operations Agent. OPC_NODENAME es un parámetro interno y no se puede configurar. Cuando un nodo tiene varias direcciones IP o nombres de host, puede sobrescribir la configuración predeterminada estableciendo OPC_IP_ADDRESS en una dirección IP específica. El nombre de host asociado a la dirección IP especificada se asigna a OPC_NODENAME. Si la dirección IP no está asociada a un nombre de host, puede configurar OPC_NAMESRV_LOCAL_NAME en un nombre de host específico. Puede usar la variable LOCAL_NODE_NAME, disponible en el espacio de nombres `xpl.net`, para sobrescribir el nombre del nodo local del sistema. Los valores definidos de OPC_NODENAME, OPC_IP_ADDRESS y LOCAL_NODE_NAME se usan para determinar varias variables de la directiva. Las variables de mensajes son las siguientes:

- <$MSG_GEN_NODE>: Devuelve la dirección IP del nodo que envía el mensaje.
- <$MSG_GEN_NODE_NAME>: Devuelve el nombre de host del nodo que envía el mensaje.
- <$MSG_NODE>: Devuelve la dirección IP del nodo en el que tuvo lugar el evento original.
- <$MSG_NODE_NAME>: Devuelve el nombre de host del nodo en el que tuvo lugar el evento original.

Nota: Los parámetros OPC_NODENAME, OPC_IP_ADDRESS y OPC_ NAMESRV_LOCAL_NAME están disponibles en el espacio de nombres `eaagt`.

Estos son algunos ejemplos:

Ejemplo 1:

A continuación se indican las direcciones IP y nombres de host en un entorno de varias interfaces de red:

```
IP1  abc.test.com  abc

IP2  xyz.test.com  xyz
```

donde IP1 e IP2 son las dos direcciones IP diferentes.

abc.test.com y xyz.test.com son los nombre de dominio completo.

abc y xyz son los nombres de host.

xyz es el nombre de host del sistema local.

Los dos parámetros OPC_NODENAME, <$MSG_GEN_NODE_NAME> establecen automáticamente el valor del nombre de dominio completo en xyz.test.com y el parámetro <$MSG_GEN_NODE> establece la dirección IP en IP2. Estas configuraciones ocurren de manera predeterminada porque xyz es el nombre de host del sistema local.

Elija configurar una dirección IP (como IP1) mediante el comando siguiente:

ovconfchg –ns eaagt –set OPC_IP_ADDRESS *IP1*

Como resultado de esta configuración, los parámetros OPC_NODENAME y <$MSG_GEN_NODE_NAME> establecerán el valor de FQDN en abc.test.com. <$MSG_GEN_NODE> establecerá la dirección IP en IP1.

Ejemplo 2:

A continuación se indican las direcciones IP y nombres de host en un entorno de varias interfaces de red:

```
IP1  xyz.test.com  xyz

IP2  xyz.test.com  xyz
```

donde IP1 e IP2 son las dos direcciones IP diferentes.

xyz.test.com es el nombre de dominio completo.

xyz es el nombre de host del sistema local.

El parámetro <$MSG_GEN_NODE> establecerá automáticamente la dirección IP en IP1. Esta configuración tiene lugar de manera predeterminada porque xyz es el nombre de host del sistema local e IP1 es la primera dirección IP (según el orden de direcciones IP) que se asocia a xyz.

Elija configurar una dirección IP diferente (como IP2) mediante el comando siguiente:

ovconfchg –ns eaagt –set OPC_IP_ADDRESS *IP2*

Como resultado de esta configuración, <$MSG_GEN_NODE> establecerá la dirección IP en IP2.

Ejemplo 3:

En un entorno de varias interfaces de red, un sistema IPv6 de Windows tiene cuatro direcciones IP configuradas en el orden siguiente:

```
IP1

IP2

IP3

IP4
```

Las direcciones IP están asociadas a nombres de host en el orden siguiente, IP1 no está asociada a un nombre de host.

```
IP3  xyz.test.com  xyz

IP2  xyz.test.com  xyz

IP4  xyz.test.com  xyz
```

De manera predeterminada, el parámetro <$MSG_GEN_NODE> establece automáticamente la dirección IP en IP2 porque IP2 es la primera dirección IP configurada (según el orden de direcciones IP) asociada al nombre del sistema local.

Elija configurar una dirección IP diferente mediante el comando siguiente:

ovconfchg –ns eaagt –set OPC_IP_ADDRESS <*dirección_IP*>

Como resultado de esta configuración, <$MSG_GEN_NODE> se establecerá en la dirección IP específica.

Ejemplo 4:

A continuación se indican las direcciones IP y nombres de host en un entorno de una única interfaz de red:

```
IP1  abc.test.com  abc

IP1  xyz.test.com  xyz
```

donde IP1 es la dirección IP.

abc.test.com y xyz.test.com son los nombre de dominio completo.

abc y xyz son los nombres de host.

xyz es el nombre de host del sistema local.

De manera predeterminada, los dos parámetros OPC_NODENAME y <$MSG_GEN_NODE_NAME> establecen automáticamente el valor de nombre de dominio completo en xyz.test.com porque xyz es el nombre de host del sistema local.

Puede elegir configurar la dirección IP IP1 mediante el siguiente comando:

ovconfchg –ns eaagt –set OPC_IP_ADDRESS *IP1*

Como resultado de esta configuración, OPC_NODENAME y <$MSG_GEN_NODE_ NAME> establecerá el valor de nombre de dominio completo en abc.test.com. Esta configuración tiene lugar porque abc.test.com es el primer nombre de dominio completo asociado con la dirección IP IP1.

Ejemplo 5:

A continuación se indican las direcciones IP en un entorno de varias interfaces de red:

```
IP1

IP2  xyz.test.com  xyz
```

donde IP1 e IP2 son las dos direcciones IP diferentes.

IP1 no está asociada a un nombre de host.

xyz.test.com es el nombre de dominio completo.

xyz es el nombre de host del sistema local.

De manera predeterminada, los dos parámetros OPC_NODENAME, <$MSG_GEN_ NODE_NAME> establecen automáticamente el valor del nombre de dominio completo en xyz.test.com y el parámetro <$MSG_GEN_NODE> establece la dirección IP en IP2.

Elija configurar una dirección IP (como IP1) mediante el comando siguiente:

ovconfchg –ns eaagt –set OPC_IP_ADDRESS *IP1*

Como resultado de esta configuración, OPC_IP_ADDRESS se establece en IP1 y no está asociada a un nombre de host. Puede configurar el nombre de host correspondiente a OPC_IP_ADDRESS usando el comando siguiente:

ovconfchg –ns eaagt –set OPC_NAMESRV_LOCAL_NAME*<nombre_host>*

donde *<nombre_host>* es cualquier nombre como abc.test.com. Al ejecutar ambos comandos, los dos parámetros OPC_NODENAME y <$MSG_GEN_NODE_NAME> establecerán el valor del nombre de dominio completo en abc.test.com.

Nota: El parámetro OPC_IP_ADDRESS establecido en un entorno NAT se comporta de la misma forma tal como se ha descrito en los ejemplos anteriores.

Ejemplo 6:

A continuación se indican las direcciones IP y nombres de host en un entorno de una única interfaz de red:

```
IP1  abc.test.com  abc xyz

IP1  xyz.test.com  xyz
```

donde IP1 es la dirección IP.

abc.test.com y xyz.test.com son los nombre de dominio completo.

abc y xyz son los nombres de host.

xyz es el nombre de host del sistema local.

De manera predeterminada, los dos parámetros OPC_NODENAME y <$MSG_GEN_NODE_NAME> establecerán automáticamente el valor del nombre de dominio completo en abc.test.com. Si el evento externo se origina en el nodo con el nombre xyz, el parámetro <$MSG_NODE_NAME> se establece automáticamente en xyz.

Puede elegir configurar OPC_NODENAME, <$MSG_GEN_NODE_NAME> y también <$MSG_NODE_NAME> en el valor del nombre de dominio completo xyz.test.com mediante la ejecución de una de las opciones siguientes:

- Intercambie el orden de las entradas en el archivo de host.
- Suprima el alias **xyz** de la primera entrada.

Ejemplo 7:

A continuación se indican las direcciones IP y nombres de host de los dos sistemas:

```
IP1  abc.test.com  abc

IP2  xyz.test.com  xyz
```

donde IP1 y abc son las direcciones IP y el nombre de host del sistema remoto.

IP1 y xyz son las direcciones IP y el nombre de host del sistema local.

abc.test.com y xyz.test.com son los nombre de dominio completo.

Cuando un evento se origina en IP1 (por ejemplo, una captura se origina desde el nodo remoto abc), las configuraciones siguientes se producen de forma predeterminada:

- El parámetro <$MSG_NODE> se establece en IP1 cuando <$MSG_NODE> devuelve la dirección IP del nodo en el que tuvo lugar el evento original.
- El parámetro <$MSG_NODE_NAME> se establece en abc cuando <$MSG_NODE_NAME> devuelve el nombre de host del nodo en el que tuvo lugar el evento original.
- El parámetro <$MSG_GEN_NODE> se establece en IP2 cuando <$MSG_GEN_NODE> devuelve la dirección IP del nodo que envía el mensaje.

- El parámetro <$MSG_GEN_NODE_NAME> se establece en xyz cuando <$MSG_GEN_NODE_NAME> devuelve el nombre de host del nodo que envía el mensaje.

Cuando la variable LOCAL_NODE_NAME está configurada, HPE Operations Agent utiliza LOCAL_NODE_NAME como nombre de host del sistema local.

Para configurar otro nombre de host local (como xyz), use el siguiente comando:

```
ovconfchg -ns xpl.net -set LOCAL_NODE_NAME xyz
```

Como resultado de esta configuración, los parámetros OPC_NODENAME y <$MSG_GEN_NODE_NAME> establecerán el valor de la variable LOCAL_NODE_NAME en xyz.

Capítulo 12: Registro y rastreo

Puede diagnosticar y solucionar problemas en HPE Operations Agent mediante los mecanismos de registro y rastreo. HPE 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 HPE para su análisis.

Registro

HPE 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 HP-UX/Linux/Solaris:

`/var/opt/OV/log`

Además, HPE Operations Agent agrega detalles de estado de Componente Performance Collection y coda a los siguientes archivos:

En Windows:

- `%ovdatadir%\log\System.txt`
- `%ovdatadir%\status.perfalarm`
- `%ovdatadir%\status.ttd`
- `%ovdatadir%\status.mi`
- `%ovdatadir%\status.perfd-`*<puerto>*
- `%ovdatadir%\hpcs\hpcstrace.log`

Sugerencia: 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 HP-UX/Linux/Solaris:

- `/var/opt/OV/log/System.txt`
- `/var/opt/perf/status.perfalarm`
- `/var/opt/perf/status.ttd`
- `/var/opt/perf/status.mi`
- `/var/opt/perf/status.perfd`
- `/var/opt/OV/hpcs/hpcstrace.log`

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 HPE Operations Agent para restringir el tamaño del archivo `System.txt`.

Para modificar la directiva de registro predeterminada, siga estos pasos:

1. Inicie sesión en el nodo.
2. Vaya a la ubicación siguiente:

   En Windows:

   `%ovdatadir%conf\xpl\log`

   En HP-UX/Linux/Solaris:

   `/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 Operations Monitoring con los siguientes comandos:
   a. `ovc -kill`
   b. `ovc -start`

Rastreo

Antes de iniciar el rastreo de una aplicación de HPE 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).

Los argumentos y opciones de línea de comandos para el rastreo se enuncian en la tabla siguiente.

Opciones de la línea de comandos

Opciones de comando	Descripción
`-host`	Especifica el nombre de host de la máquina a la que conectarse. Sintaxis: `-host <nombre_de_host>`
`-off`	Desactive todos los rastreos.
`-cf|-configuration`	Especifica el nombre del archivo de configuración para cambios dinámicos. Sintaxis: `-cf <nombre_de_archivo_de_configuración>`
`-vc|-viewconfig`	Muestra la configuración de rastreo de todas las aplicaciones.
`-app|-application`	Especifica la lista de nombres de aplicación que se van a configurar.
`-cm|-component`	Especifica la lista de nombres de componentes que se van a configurar.
`-sink`	Especifica el nombre de archivo de salida de rastreo. Sintaxis: `-sink <nombre_de_archivo>`
`-gc|-generate_configuration`	Especifica el nombre del archivo de configuración que se va a generar. Sintaxis: `-gc <nombre_de_archivo>`
`-rd|-resetdefault`	Restablece todos los archivos `.ini` a su contenido predeterminado.
`-h|-help`	Muestra la sintaxis de la línea de comandos.
`-version`	Muestra el número de versión de la herramienta.

Antes de iniciar el rastreo de un proceso de HPE Operations Agent, realice las siguientes tareas:

1. Identificación de la aplicación
2. Establecimiento del tipo de rastreo
3. *Opcional*. Creación del archivo de configuración

Identificación de la aplicación

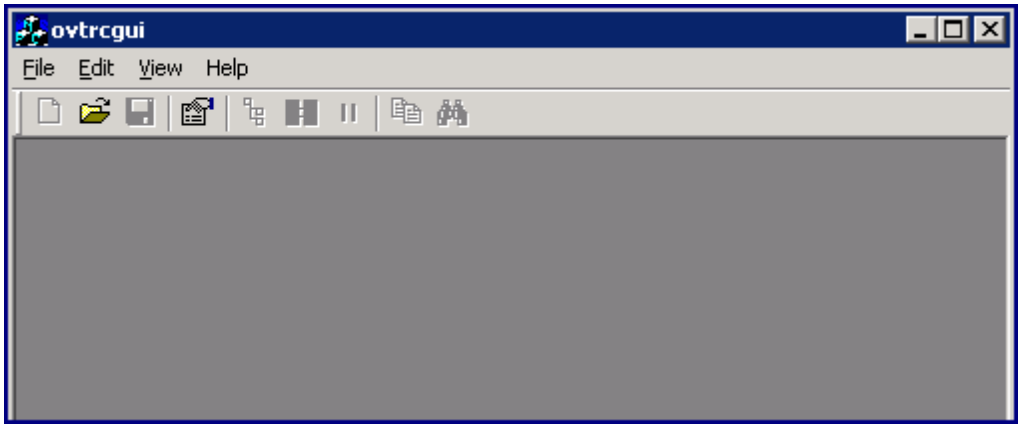
En el sistema administrado, identifique las aplicaciones de HPE Software que desee rastrear. Use la opción `ovtrccfg -vc` para ver los nombres de todas las aplicaciones con el rastreo activado y los componentes y categorías definidos para cada una de ellas.

Si lo prefiere, puede usar la utilidad ovtrcgui para ver la lista de aplicaciones con el rastreo activado.

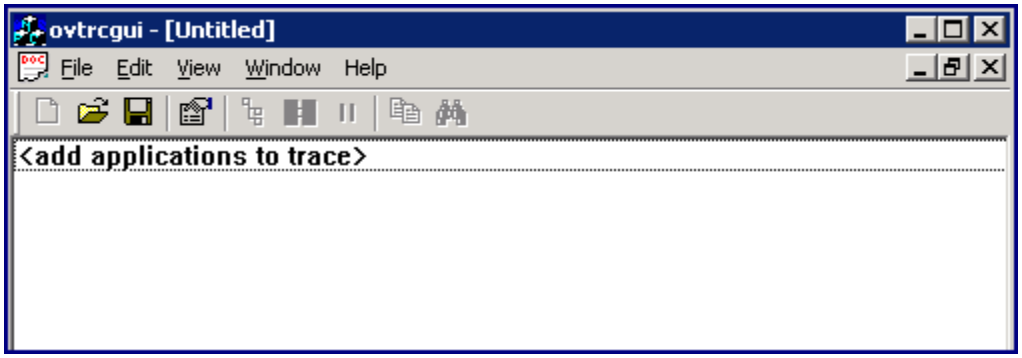
Nota: La utilidad `ovrtrcgui` solo se admite en los sistemas Windows.

Para usar la utilidad `ovtrcgui` para ver la lista de aplicaciones con el rastreo activado, siga estos pasos.

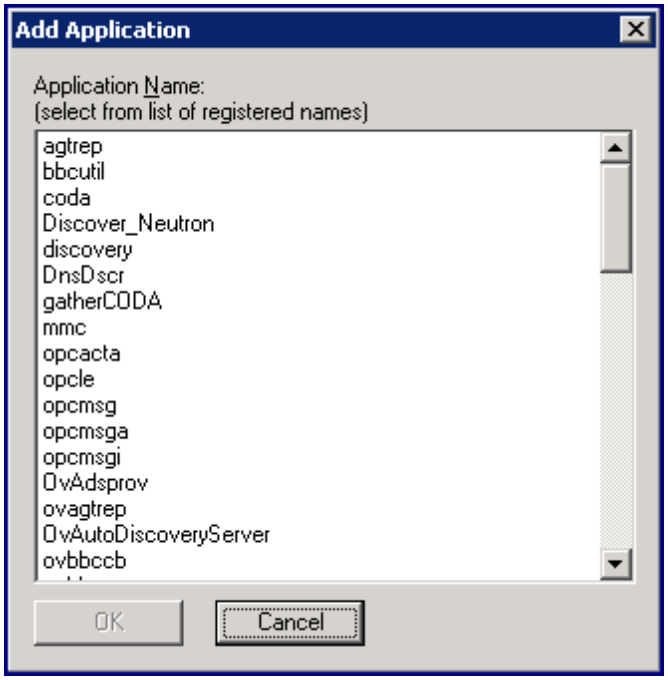
1. Ejecute el archivo `ovtrcgui.exe` en el directorio *%OvInstallDir%*`\support`. Se abre la ventana de `ovtrcgui`.



2. En la ventana de `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 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`.

     Sugerencia: 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: "<nomre_aplicación>"

SINK: File "<nombre_archivo>" "maxfiles=[1..100];maxsize=
[0..1000];"
```

`TRACE: "<nombre_componente>" "<nombre_categoría>" <lista_palabras_clave>`

Cada línea de la sintaxis se describe detalladamente en las siguientes secciones.

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*.

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).

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. Habilite dinámicamente el rastreo para las aplicaciones específicas mencionadas en el archivo de configuración mediante el comando `ovtrccfg`.
   `/opt/OV/support/ovtrccfg -cf <nombre_archivo_configuración>`

   donde *<nombre_archivo_configuración>* es el nombre del archivo de configuración del rastreo creado en la sección anterior *Creación del archivo de configuración*.

   Nota: Si no desea usar un archivo de configuración del rastreo, puede habilitar el rastreo con el siguiente comando:

   `/opt/OV/support/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:
     `/opt/OV/support/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:
     `/opt/OV/support/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:
     `/opt/OV/support/ovtrcmon -fromfile /opt/OV/bin/trace1.trc -short > /tmp/traces.trc`
5. Para detener o deshabilitar el rastreo mediante `ovtrccfg`, ejecute el siguiente comando:
   `/opt/OV/support/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 HPE 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`).

   Por ejemplo:

```
SINK: File "coda.trc" "force=0;maxfiles=10;maxsize=10;"
root@/var/opt/OV/tmp/trc->lrt
total 93092
-rw-r----- 1 root bin 10475620 Aug 12 15:30 coda_00013.trc
-rw-r----- 1 root bin 10475528 Aug 12 15:30 coda_00014.trc
-rw-r----- 1 root bin 10475528 Aug 12 15:30 coda_00015.trc
-rw-r----- 1 root bin 10475528 Aug 12 15:30 coda_00016.trc
-rw-r----- 1 root bin 10475528 Aug 12 15:30 coda_00017.trc
-rw-r----- 1 root bin 10475528 Aug 12 15:30 coda_00018.trc
-rw-r----- 1 root bin 10475528 Aug 12 15:30 coda_00019.trc
-rw-r----- 1 root bin 10475528 Aug 12 15:30 coda_00020.trc
```

```
-rw-r----- 1 root bin 10475528 Aug 12 15:31 coda_00021.trc
-rw-r----- 1 root bin  1027187 Aug 12 15:31 coda_00022.trc
root@/var/opt/OV/tmp/trc->
```

Habilitación del rastreo y visualización de mensajes de rastreo con la GUI de rastreo

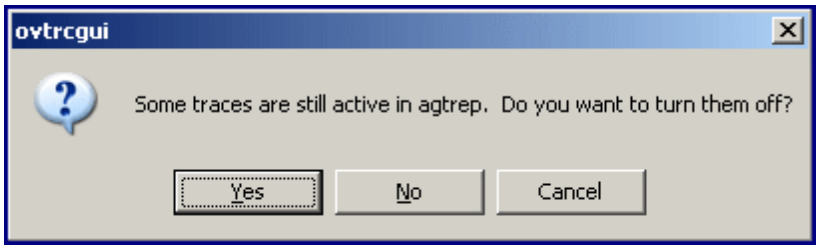
En los nodos de Windows, puede usar la utilidad `ovtrcgui` para configurar el rastreo y ver los mensajes del rastreo.

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 el paso 1 al 6 de "Uso de la GUI de rastreo" en la página 273.
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:



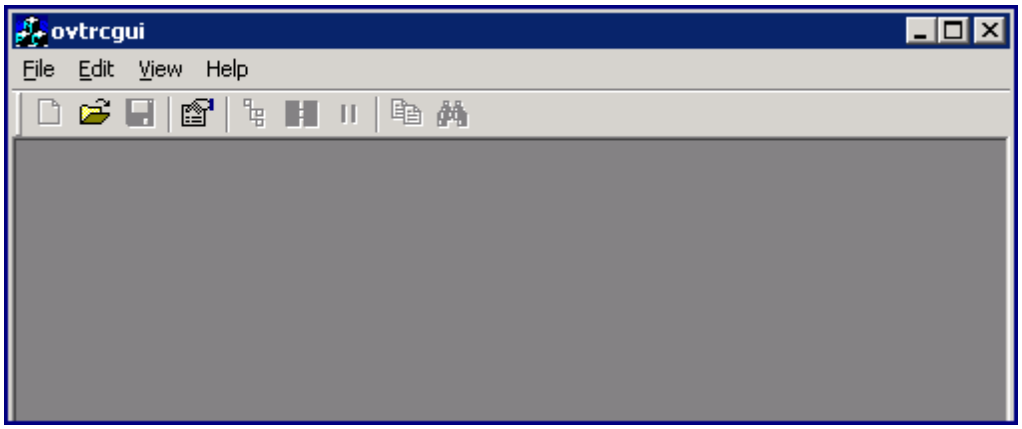
4. Haga clic en **No**. Si hace clic en **Yes**, la utilidad `ovtrcgui` deshabilita inmediatamente el mecanismo de rastreo.

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 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**.

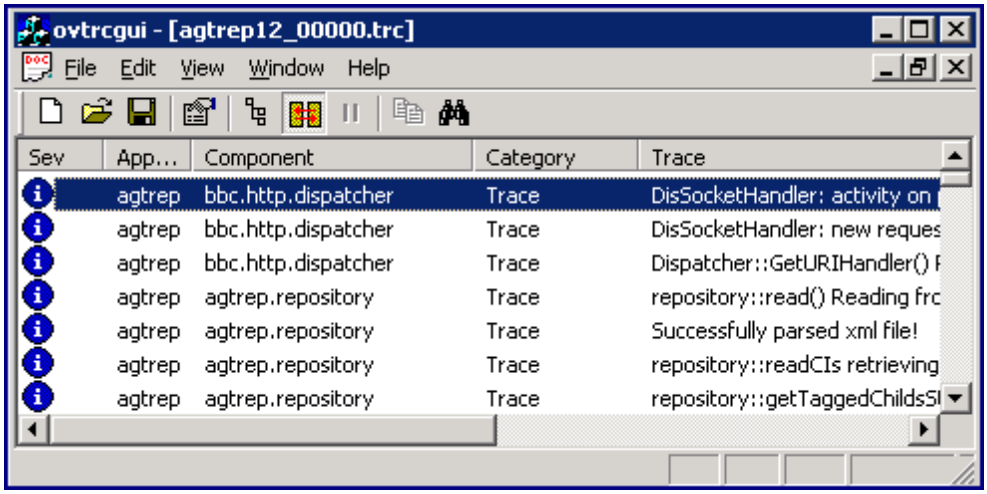
Visualización de mensajes de rastreo

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 de ovtrcgui.

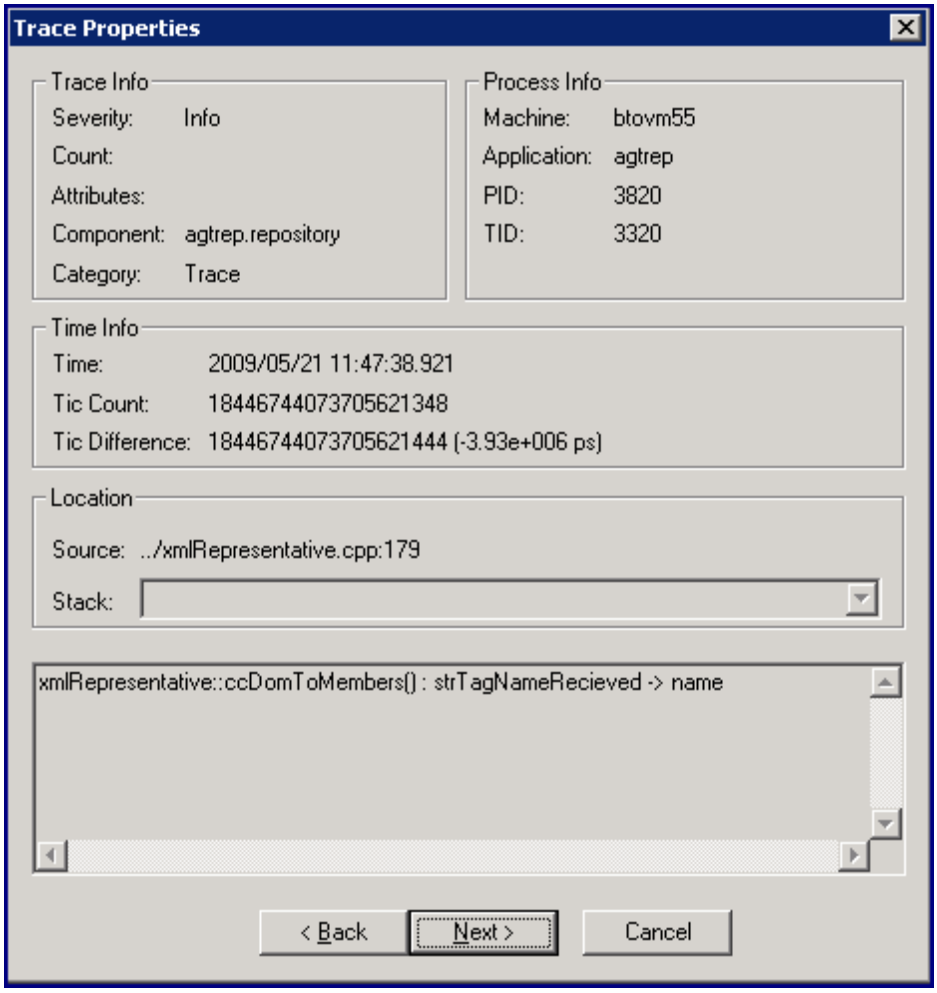


2. Haga clic en **File** → **Open**. Se abrirá el cuadro de diálogo Open.
3. Desplácese a la ubicación del archivo de resultados del rastreo, seleccione el archivo `.trc` y haga clic en **Open**. La utilidad ovtrcgui para muestra el contenido del archivo `.trc`.



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:* El nombre del componente que emite el mensaje de rastreo.
  - *Category:* El nombre arbitrario asignado por la aplicación rastreada.
- Process Info:
  - *Machine:* El nombre del host del nodo.
  - *Application:* Nombre de la aplicación rastreada.
  - *PID:* El ID del proceso de la aplicación rastreada.
  - *TID:* El ID del subproceso de la aplicación rastreada.

- Time Info:
  - *Time:* La fecha y hora equivalente local del mensaje de rastreo.
  - *Tic count:* El tiempo transcurrido de alta resolución.
  - *Tic difference:*
- Location
  - *Source:* El 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.

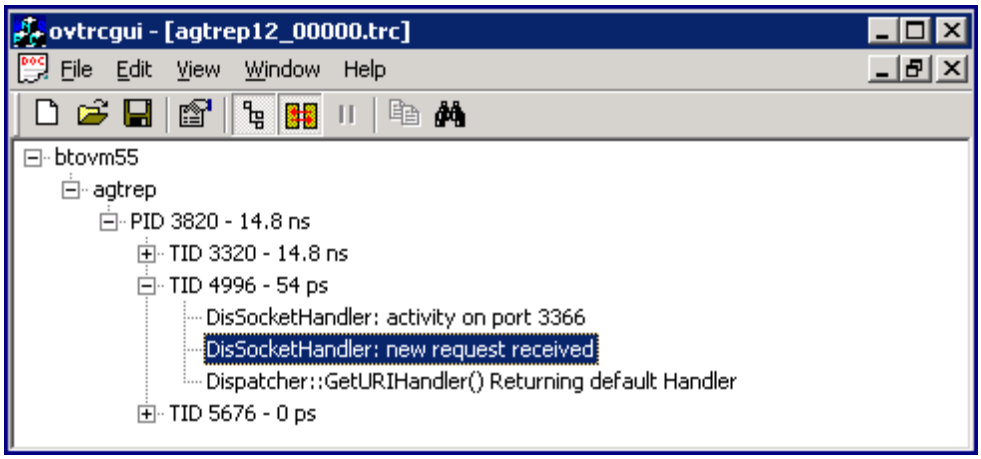
La vista de lista de rastreos presenta cada mensaje de rastreo con ayuda de las siguientes columnas:

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:	
	Info	
	Advertencia	
	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.

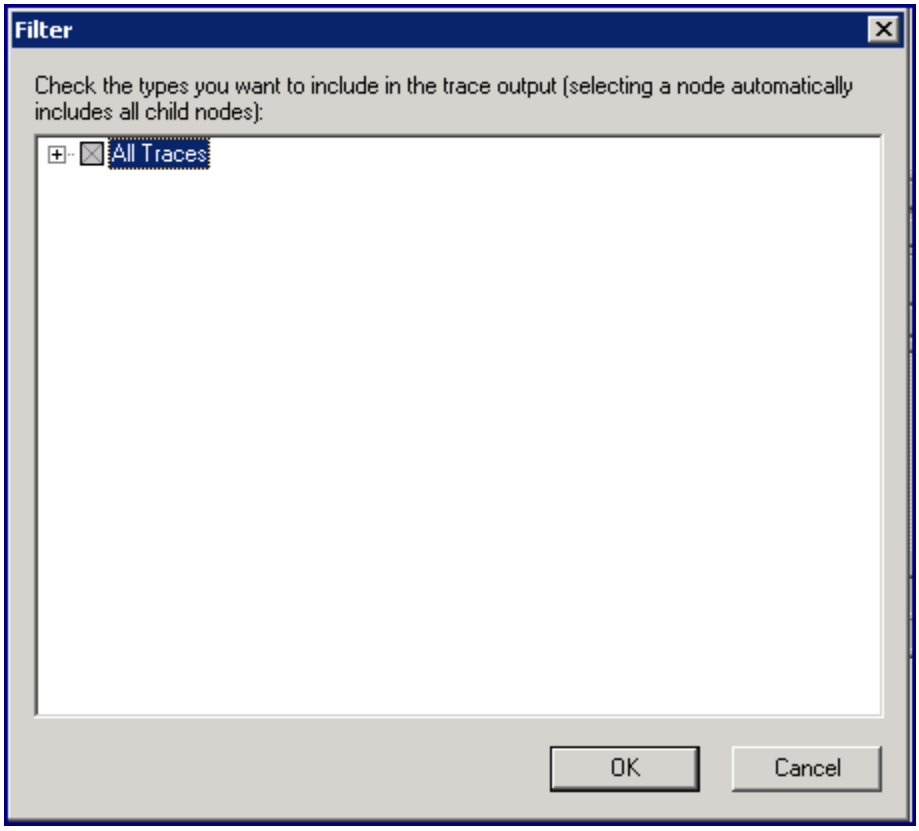


Puede ampliar los Id. de proceso y los Id. de subprocesos para ver los mensajes de rastreo. Para volver a la vista de la lista de rastreo, haga clic en  .

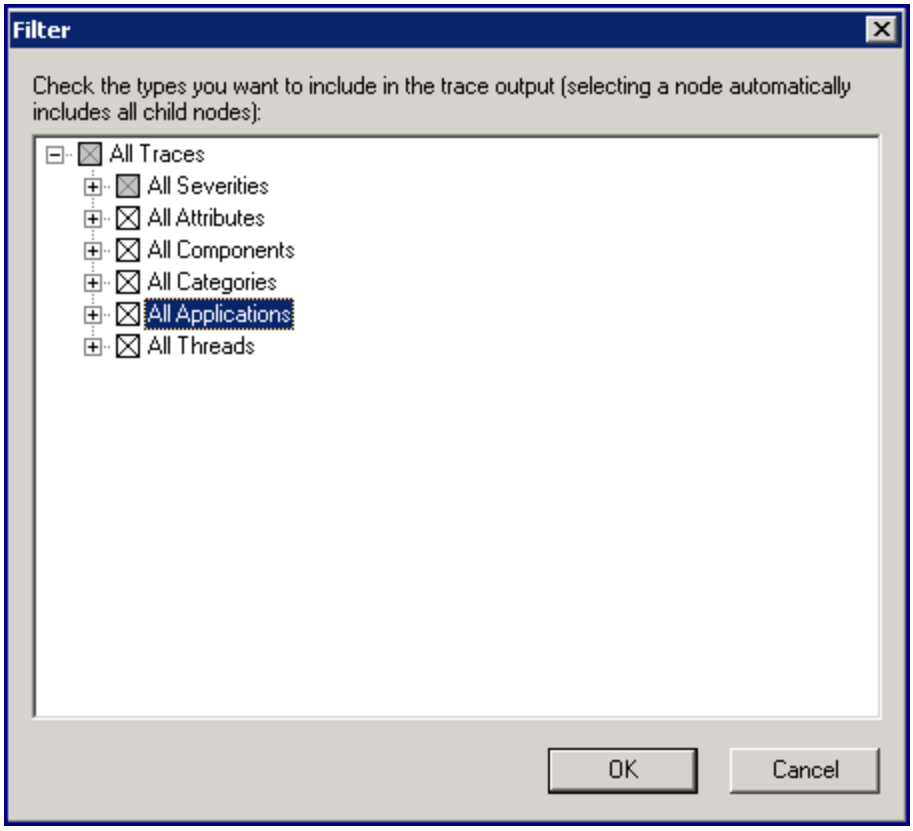
Filtrado de rastreos

La utilidad `ovtrcgui` 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 ovtrcgui. Para filtrar los mensajes de rastreo disponibles, siga estos pasos:

1. En la consola `ovtrcgui`, haga clic en **View** → **Filter**. Se abrirá el cuadro de diálogo Filter.



2. Amplíe **All Traces**. En el cuadro de diálogo se enumeran todos los parámetros de filtrado en forma de árbol.

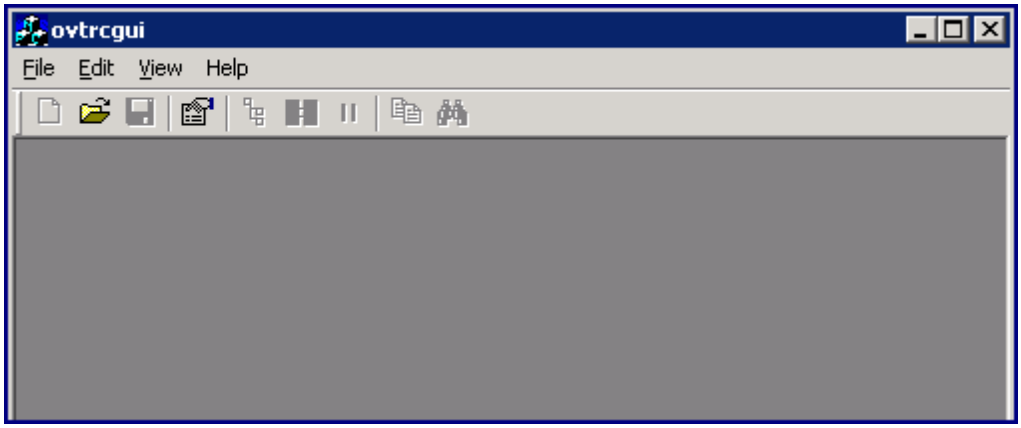


3. Amplíe los parámetros para realizar selecciones para filtrar los mensajes de rastreo.
4. Haga clic en **OK**. Sólo puede ver los mensajes filtrados en la consola `ovtrcgui`.

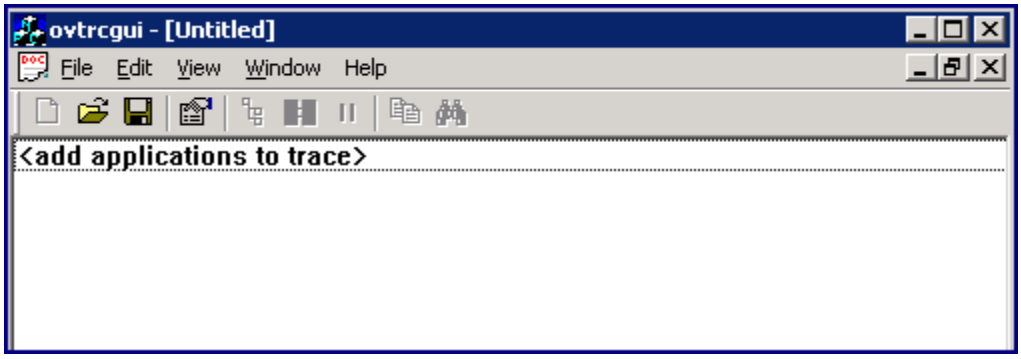
Uso de la GUI de rastreo

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:

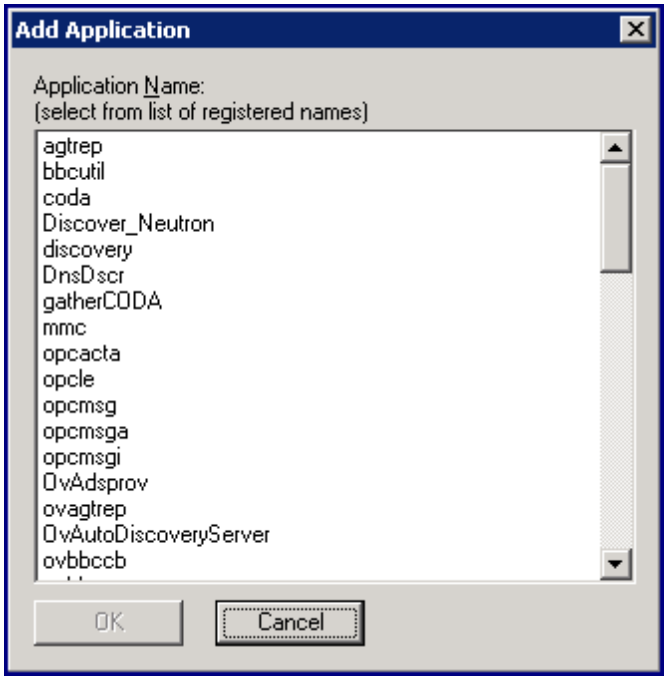
1. Ejecute el archivo `ovtrcgui.exe` en el directorio *%OvInstallDir%*`\support`. Se abre la ventana de ovtrcgui.



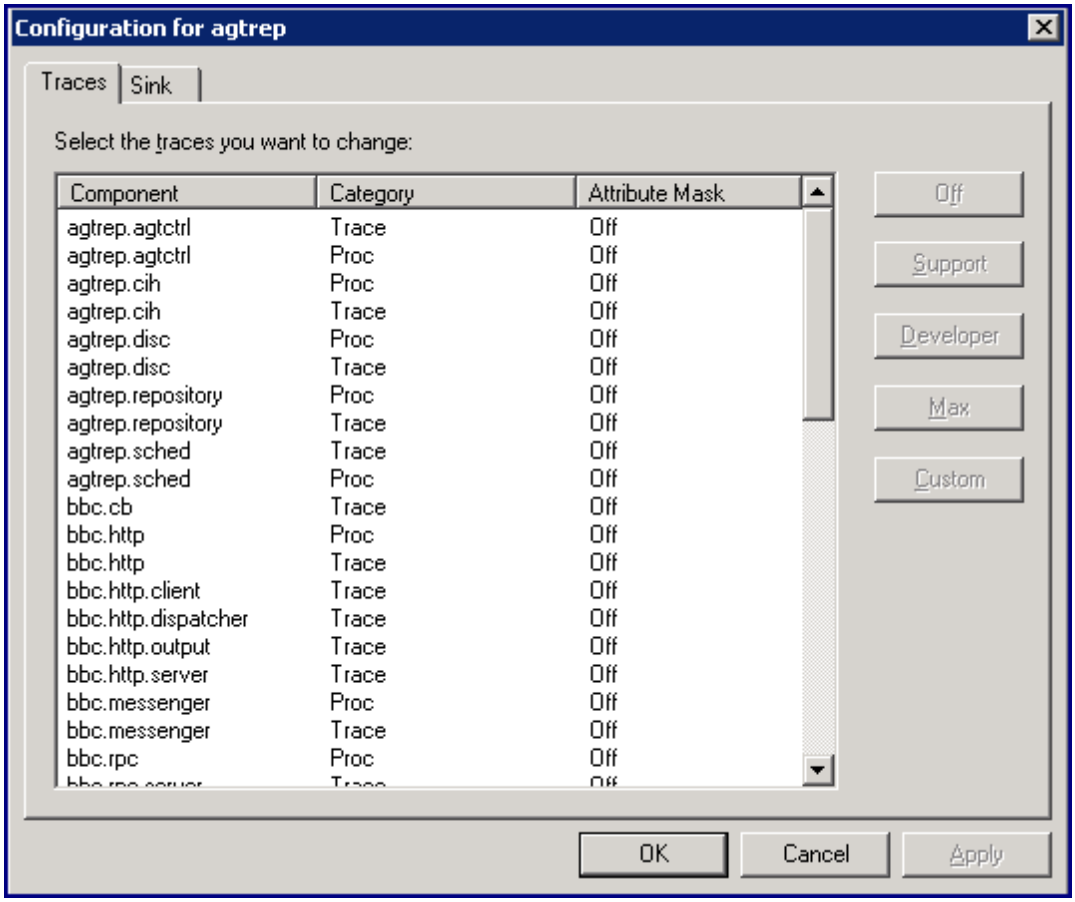
2. En la ventana de ovtrcgui, haga clic en **File** → **New** →**Trace Configuration**. Se abre un nuevo editor de configuración de rastreo.



3. 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.

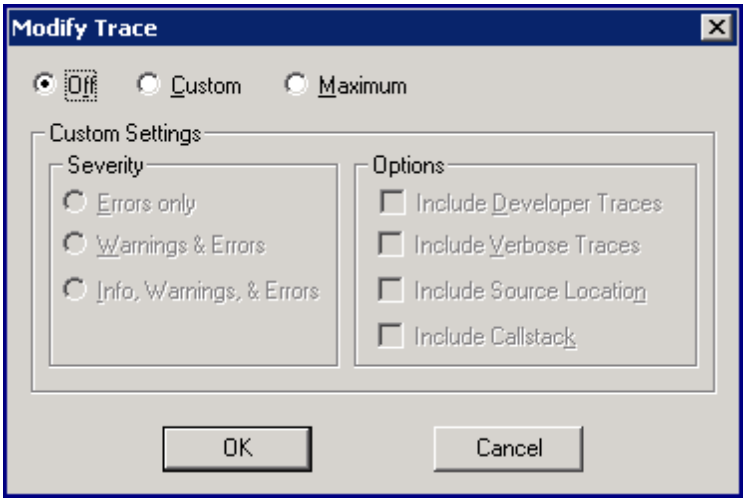


4. Seleccione la aplicación que desea rastrear y haga clic en **OK**. Se abre la ventana Configuration for <*aplicación*>.



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 Personalizar, aparece la ventana Modificar seguimiento.



En la ventana Modify Trace, seleccione las opciones personalizadas, los niveles de rastreo y las opciones que desee y haga clic en **OK**.

Sugerencia: En la ventana Configuration for *<aplicación>*, puede hacer clic en **Off** para deshabilitar 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`.

   Sugerencia: Especifique la ruta completa del archivo `.trc`.

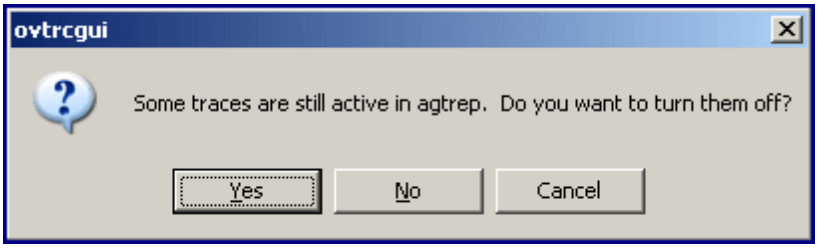
9. Especifique el número de archivos históricos en la lista desplegable (consulte maxfiles).
10. Especifique el tamaño de archivo máximo en la lista desplegable (consulte maxsize).
11. Haga clic en **Apply**.
12. Haga clic en **OK**.

    Nota: 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.

    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**.

14. 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.

Parte IV: Registro de datos personalizados

HPE Operations Agent le permite registrar los datos personalizados en el Almacén de datos de métricas junto con las clases de métricas de rendimiento personalizadas. Puede usar el envío de API o DSI para registrar datos personalizados en el Almacén de datos de métricas.

El proceso de registrar datos personalizados en el Almacén de datos de métricas se simplifica con el uso de API que permiten ejecutar scripts basadas en Perl. Puede usar las API para registrar datos de varias instancias y tipos de datos de 64 bits. Para obtener más información sobre las API, consulte Uso de la interfaz de programación de aplicaciones.

Para enviar datos personalizados con DSI, debe crear un archivo de especificación de clase que define el modelo para registrar datos y después registrar los datos en el Almacén de datos de métricas. No puede registrar tipos de datos de 64 bits y datos de múltiples instancias con DSI. Para obtener más información, consulte Descripción general de la integración de orígenes de datos.

Nota: Se recomienda que utilice las API para registrar datos personalizados en el Almacén de datos de métricas.

Capítulo 13: Uso de la interfaz de programación de aplicaciones Perl para enviar datos personalizados

El proceso de registro de datos personalizados se ha simplificado con el uso de API que permiten ejecutar scripts basados en Perl. Puede usar API para registrar perfectamente datos personalizados en el Almacén de datos de métricas. Con las API puede registrar tipos de datos de 64 bits y datos de varias instancias en el almacén de datos.

Nota: Las API se pueden usar para registrar los datos en el Almacén de datos de métricas solo si se está ejecutando el proceso **oacore**.

HPE Operations Agent 12.xx proporciona las siguientes API para registrar los datos personalizados en el Almacén de datos de métricas:

Tipo de API	Tipo de resumen	Tipo de datos	Cómo elegir un tipo de datos
AddCounter	Da el último valor del intervalo resumido.	Entero	Use la métrica Counter para enviar un recuento acumulado de la actividad, como los tiempos de CPU, las E/S físicas, la paginación, los recuentos de paquetes de red y similares, al almacén de datos.
AddGauge	Da la media de todos los valores del intervalo de resumen.	Entero, real	Use la métrica Gauge para enviar el valor puntual que aparece en el momento de la observación, como la cola de ejecución, el número de usuarios, el uso del espacio del sistema de archivos y

			similares, al almacén de datos.
AddAttribute	Da los datos que no cambian con frecuencia. No se puede realizar resumen. **Nota:** El historial de los datos de los atributos no se guarda en el almacén de datos. Siempre que se cambia un atributo, se sustituye en el almacén de datos.	Entero, real, cadena	Use la métrica Attribute para enviar definiciones estáticas o un valor, como el nombre o la versión del sistema operativo, la memoria física, la velocidad de reloj de la CPU y similares, al almacén de datos.

Nota:

- Asegúrese de que no envía una métrica utilizando otros tipos de API.

  En la siguiente instancia:

  ```
  AddCounter("Website:Global:UserLogins", "www.abcdefg.com", "Cumulative User Logins", 6000);
  AddGuage("Website:Global:UserLogins", "www.abcdefg.com", "Cumulative User Logins", 5000);
  ```

  Si se envía la métrica 'UserLogins' utilizando los tipos de API Counter y Gauge, se obtiene un resultado incorrecto.
- Después de enviar una métrica no es posible cambiar el tipo de métrica.

Por ejemplo: Después de enviar una métrica como Counter no se puede cambiar a Gauge o Attribute.

En este documento se usan los siguientes términos:

- El objeto MetricObservation contiene los datos que se van a registrar en el almacén de datos, junto con la marca de fecha y hora de la recopilación.

- MetricObservationList contiene varios objetos MetricObservation.
- OAAccess es un cliente que se conecta al almacén de datos.
- La API SubmitObservations se usa para enviar MetricObservationList al Almacén de datos de métricas.

  SubmitObservations devuelve uno de los siguientes valores:
  - 0 - MetricObservationList se envió correctamente al Almacén de datos de métricas
  - 1 - Error en el envío de MetricObservationList
  - 2 - El proceso **oacore** no se está ejecutando

Envío de datos personalizados

Sintaxis

```
<AddMethod>(<nombre de origen de datos>:<nombre de clase>:<nombre de métrica>, <identificador de instancia>, <etiqueta de la métrica>,<valor de métrica>)
```

Argumentos de API

La siguiente tabla muestra los argumentos:

Argumento	Descripción
Nombre completo de métrica	Especifica el nombre completo de la métrica. Incluye el nombre del origen de datos, el nombre de la clase y el nombre de la métrica. <nombre de origen de datos>:<nombre de clase>:<nombre de métrica>
Identificador de instancia	Especifica la instancia de la clase. Puede ser un número entero, un valor real o un valor de cadena.
Etiqueta	Especifica la etiqueta de la métrica.
Valor	Especifica el valor de la métrica que se va a enviar. **Nota:**

	En los métodos AddGauge, Valor puede ser un número entero o un valor real. En el método AddAttribute , Valor puede ser un número entero, un valor real o un valor de cadena.

Por ejemplo:

AddGauge("MySQL:Global:MySQLMemUtil", "MyHost:3306", "MySQL Total CPU utilization percentage", 20.12);

Donde:

AddMethod	AddGauge
Nombre de origen de datos	MySQL
Nombre de clase	Global
Nombre de métrica	MySQLMemUtil
Identificador de instancia	MyHost:3306
Etiqueta de la métrica.	MySQL Porcentaje total de utilización de la CPU
Valor de la métrica	20.12

Envío de datos personalizados a Almacén de datos de métricas

Para enviar datos a Almacén de datos de métricas, cree objetos MetricObservation. Añada los objetos MetricObservation a MetricObservationList.

Nota: Se pueden añadir tantos objetos MetricObservation como se desee a MetricObservationList.

Envíe MetricObservationList a Almacén de datos de métricas con la API SubmitObservations.

Nota: Para invocar a la API SubmitObservations, es preciso crear un objeto

OAAccess.

La siguiente tabla muestra los pasos necesarios para enviar datos a Almacén de datos de métricas, junto con un ejemplo:

No.	Pasos para enviar datos personalizados a Almacén de datos de métricas	Por ejemplo
1.	Cree OAAccess.	$access = oaperlapi::OAAccess->new();
2.	Cree un objeto MetricObservationList.	$mol = oaperlapi::MetricObservationList->new();
3.	Cree un objeto MetricObservation con una marca de fecha y hora. **Nota:** Se pueden crear varios objetos MetricObservation con distintas marcas de fecha y hora. Si la hora es cero, se toma la hora actual. En todas las clases de datos que envíe, asegúrese de que las marcas de fecha y hora de MetricObservations están en orden creciente.	$interval = time; $mo = oaperlapi::MetricObservation->new($interval);
4.	Agregue valores al objeto MetricObservation a través de métodos como AddGauge o AddAttribute.	$mo->AddAttribute ("MySql:Global:MySQLVersion", "MyHost:3306", "MySQL version", "5.6.0"); $mo->AddGauge ("MySql:Global:MySQLMemUtil", "MyHost:3306", "MySQL Total CPU utilization percentage",

		20.12);
5.	Añada los objetos MetricObservation al objeto MetricObservationList.	$mol->AddObservation($mo);
6.	Envíe MetricObservationList a Almacén de datos de métricas con la API SubmitObservations.	$access->SubmitObservations ($mol);

Por ejemplo:

```
$access = oaperlapi::OAAccess->new();

$mol = oaperlapi::MetricObservationList->new();

$interval = time;

$mo = oaperlapi::MetricObservation->new($interval);

$mo->AddAttribute("MySql:Global:MySQLVersion", "MyHost:3306", "MySQL version", "5.6.0");

$mo->AddGauge("MySql:Global:MySQLMemUtil", "MyHost:3306", "MySQL Total CPU utilization percentage", 20.12);

$mol->AddObservation($mo);

$access->SubmitObservations($mol);
```

Caso de uso de envío de datos al almacén de datos mediante API

Escenario

Juan es administrador web del sitio www.abcdefg.com. Desea analizar el número de usuarios que visitan dicho sitio web desde EE.UU. Juan también quiere monitorizar el número de compras realizadas entre las 10 A.M y las 11 A.M.

Descripción

Juan desea conocer los siguientes valores:

- Número de usuarios que iniciaron sesión durante el transcurso de una hora desde EE.UU.
- Número de compras realizadas en una hora.

Actor

John: administrador web

Condición previa

El tiempo durante en que se monitorizarán los inicios de sesión y las compras es una hora.

Intervalo de tiempo	Inicios de sesión de usuario acumulados	Location	Compras en una hora
10:00 – 10:30	6000	EE.UU.	400
10:30 – 11:00	8000	EE.UU.	600

Nota:

Asegúrese de que la ruta va precedida por `<directorio_de_instalación>/nonOV/perl/a/bin`

Use los siguientes comandos para ejecutar el script siguiente

En Windows x64:

```
perl -I <dir_instalación>\support -I <dir_instalación>\nonOV\perl\a\lib\site_perl\5.16.0\MSWin32-AMD64-multi-thread <script Perl>
```

En Windows x86:

```
perl -I <dir_instalación>\support -I <dir_instalación>\nonOV\perl\a\lib\site_perl\5.16.0\MSWin32-x86-multi-thread <script Perl>
```

En HP-UX PA-RISC:

LD_PRELOAD tiene que definir la ruta de liboaperlapi.sl al ejecutar el script perl

Ejemplo:

```
LD_PRELOAD=/opt/OV/nonOV/perl/a/lib/site_perl/5.16.0/PA-RISC2.0-thread-multi/liboaperlapi.sl
/opt/OV/nonOV/perl/a/bin/perl <script Perl>
```

En UNIX:

```
perl <script Perl>
```

Para enviar los datos al almacén de datos, ejecute el siguiente script Perl:

```
use oaperlapi;

$now_string = localtime;
```

 # Paso 1: Crear el objeto de acceso

```
$access = oaperlapi::OAAccess->new();
```

 # Paso 2: Crear un objeto MetricObservationList

```
$molist = oaperlapi::MetricObservationList->new();
$obsTimestamp = 1415939400;
```

```
Nota: In this instance, $obsTimestamp represent the time when the
first record was logged into the datastore.
```

 # Paso 3: Crear un objeto MetricObservation con una marca de fecha y hora

```
$mgbl = oaperlapi::MetricObservation->new($obsTimestamp);
```

 # Paso 4: Agregar valores al objeto MetricObservation

```
$mgbl->AddAttribute("Website:Global:Location", "www.abcdefg.com",
"Location", "USA");
$mgbl->AddCounter("Website:Global:UserLogins", "www.abcdefg.com",
"Cumulative User Logins", 6000);
$mgbl->AddGauge("Website:Global:Purchases", "www.abcdefg.com", "User
Purchases in an hour", 400);
```

 # Paso 5: Agregar objetos MetricObservation a MetricObservationList.

```
$molist->AddObservation($mgbl);
# Go to the next Timestamp
$obsTimestamp += 1800;
```

 # Repita los pasos 3, 4 y 5 para crear otra observación, tal como se muestra:

```
$mgbl = oaperlapi::MetricObservation->new($obsTimestamp);
$mgbl->AddCounter("Website:Global:UserLogins", "www.abcdefg.com",
"Cumulative User Logins", 8000);
$mgbl->AddGauge("Website:Global:Purchases", "www.abcdefg.com", "User
Purchases in an hour", 600);
$molist->AddObservation($mgbl);
```

 # Paso 6: Enviar datos de forma masiva

```
$access->SubmitObservations($molist);
```

Los siguientes datos se registran en el almacén de datos:

```
===
11/14/14 10:00:00|GLOBAL_ID                    |www.abcdefg.com|
11/14/14 10:00:00|LOCATION                      |USA            |
11/14/14 10:00:00|USERLOGINS                    |           6000|
11/14/14 10:00:00|PURCHASES                     |        400.00|
===
11/14/14 10:30:00|GLOBAL_ID                    |www.abcdefg.com|
11/14/14 10:30:00|LOCATION                      |USA            |
11/14/14 10:30:00|USERLOGINS                    |           8000|
11/14/14 10:30:00|PURCHASES                     |        600.00|
```

Capítulo 14: Introducción general de Integración de Orígenes de Datos (DSI)

La característica de integración de orígenes de datos (DSI) le ayuda a registrar datos personalizados, definir alarmas y acceder a métricas de nuevos orígenes de datos distintos de las métricas registradas por el recopilador de datos **oacore** de Componente Performance Collection. La métrica puede proceder de orígenes de datos como bases de datos, monitores LAN y aplicaciones de usuarios finales.

Puede usar HP Performance Manager para ver los datos registrados por DSI, junto con las métricas de rendimiento estándar registradas por el recopilador de datos de HPE Operations Agent. También puede usar el programa Extract para exportar los datos registrados mediante DSI para mostrarlos en hojas de cálculo o paquetes de análisis similares.

Actualización a HPE Operations Agent 12.xx

Con HPE Operations Agent 12.01, los datos recopiladas por DSI se almacenan en Almacén de datos de métricas. Para cada clase de datos que se registra en el Almacén de datos de métricas, se crea un archivo de base de datos. Sin embargo, para conservar la compatibilidad con versiones anteriores, la línea de comandos sigue admitiendo el argumento del archivo de registro.

```
sdlcomp <archivo_especificación_clase> <nombre_archivo_registro>
```

El compilador de DSI **sdlcomp** no creará archivos basados en el nombre del archivo de registro; por el contrario, el nombre del archivo de registro se usa como nombre de origen de datos.

Por ejemplo:

Si `"/tmp/test_log"` o `"C:\test_log"` se proporciona como el argumento establecido del archivo de registro (en la línea de comandos), se usará test_log como nombre del origen de datos.

Antes de actualizar desde HPE Operations Agent 11.xx a 12.xx, asegúrese de que el origen de datos de DSI se agrega al `archivo datasources` ubicado en:

En HP-UX/Linux/Solaris:

```
/var/opt/OV/conf/perf
```

En Windows:

```
%ovdatadir%\conf\perf
```

Agregue la entrada siguiente al archivo de origen de datos:

```
DATASOURCE=<Datasource Name>LOGFILE=<DSI Logfile Path>
```

El modelo para registrar los datos de DSI se crea en Almacén de datos de métricas solo si el origen de datos de DSI se agregan al `archivo de orígenes de datos`.

Nota: Cualquier opción que no se admita con HPE Operations Agent 12.xx se ignora y se muestra un mensjae en el ejemplo siguiente:

Por ejemplo:

```
MAX INDEXES is obsoleted and will be ignored
INDEX BY is obsoleted and will be ignored
CAPACITY is obsoleted and will be ignored
ROLL BY is obsoleted and will be ignored
ACTION is obsoleted and will be ignored
PRECISION is obsoleted and will be ignored
```

Eliminación de los orígenes de datos personalizados registrados en la base de datos de CODA en HP Operations Agent 11.xx

Después de actualizará HPE Operations Agent 12.xx desde 11.xx, el modelo y los datos personalizados en los ficheros más antiguos se mantendrá en modo de solo lectura. Estos orígenes de datos antiguos no se eliminan si ejecuta el comando `sdlutil` (o `ddfutil`).

Puede eliminar los orígenes de datos antiguos antes o después de actualizar a HPE Operations Agent 12.xx. Después de actualizar a HPE Operations Agent 12.xx, siga estos pasos para eliminar todos los orígenes de datos personalizados registrados en la base de datos de CODA:

1. Pare el proceso de **oacore**.
2. Extraiga el conjunto de archivos de registro de CODA completo
3. Inicie el proceso **oacore**.
4. Ejecute el comando `sdlutil` para completar la limpieza.

Todos los archivos de registro de CODA situados en `%OvDataDir%\datafiles\coda*` se eliminan.

Funcionamiento de DSI

Siga estos pasos para registrar los datos de DSI en el Almacén de datos de métricas.

1. Creación de un modelo para registrar datos de DSI en el Almacén de datos de métricas Para obtener más información, consulte, Creación de un modelo para registrar datos de DSI al Almacén de datos de métricas.
   a. Cree un archivo de especificación de clase. Para obtener más información, consulte "Creación de un archivo de especificación de clase" en la página 293.
   b. Ejecute el comando siguiente para compilar el archivo de especificación de clase con el compilador de DSI, **sdlcomp**:

      ```
      sdlcomp <archivo_especificación_clase> <nombre_archivo_
      registro>
      ```

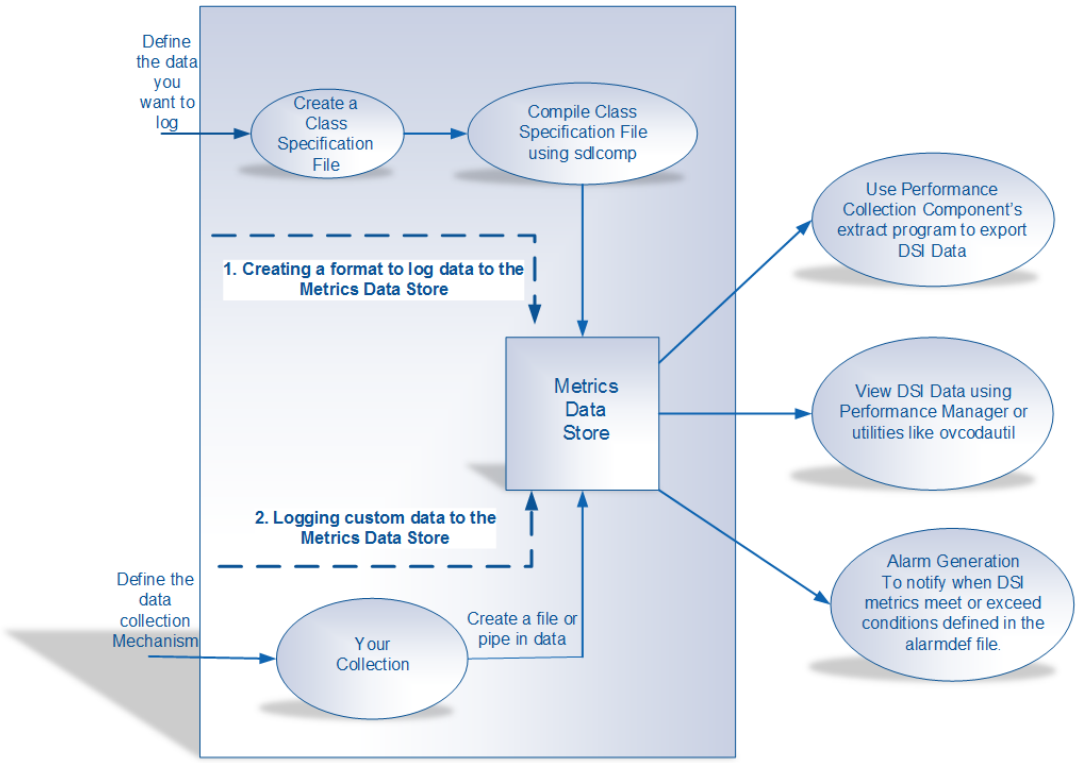
      Para obtener más información, consulte "Compilación del archivo de especificación de clase con el compilador DSI" en la página 302.

   El modelo para registrar datos (metadatos) se guarda en el Almacén de datos de métricas.
2. Registre los datos DSI en el Almacén de datos de métricas. Para obtener más información, consulte "Registro de métricas DSI en el Almacén de datos de métricas" en la página 304.

   Canalice los datos personalizados en dsilog. Dsilog registra los datos en el Almacén de datos de métricas.

El diagrama siguiente muestra cómo los datos recopilados por DSI se registran en el Almacén de datos de métricas.

Figura 17 Proceso de integración de fuentes de datos



Puede usar el programa **extract** del componente Performance Collection para exportar datos de DSI. Puede ver los datos de DSI con HP Performance Manager o utilidades como `ovcodautil`. Puede configurar las alarmas para que notifiquen cuando las métricas de DSI exceden las condiciones definidas en el archivo **`alarmdef`**.

Capítulo 15: Creación de un modelo para registrar métricas de DSI en el Almacén de datos de métricas

Antes de registrar las métricas de DSI en el Almacén de datos de métricas, debe crear y registrar el modelo para registrar métricas de DSI. Siga los pasos para registrar el modelo en el Almacén de datos de métricas.

1. Creación de un archivo de especificación de clase
2. Registro del modelo en el Almacén de datos de métricas

Creación de un archivo de especificación de clase

Por cada origen de datos de entrada será preciso crear un archivo de especificación de clase para describir el modelo de almacenamiento de los datos de entrada. Para crear el archivo, use la "Sintaxis de especificación de clase" en la página siguiente. El archivo de especificación de clase contiene:

- La descripción de clase que asigna un nombre y un ID numérico al conjunto de datos entrantes. Para obtener más información sobre la descripción de clase, consulte, "Descripción de clase" en la página 295.
- Descripción de métrica que define las métricas que se van a registrar. Las descripciones de métricas definen los nombres del elemento de datos y lo describen. Para obtener más información sobre la descripción de métricas, consulte, Descripciones de métrica.

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. 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).
- Nombres de métrica

A continuación se muestra un ejemplo de una especificación de clase:

```
CLASS VMSTAT_STATS = 10001
LABEL "VMSTAT data"
```

```
RECORDS PER HOUR 120
;

METRICS
RUN_Q_PROCS  = 106
LABEL   "Procs in run q"
;

BLOCKED_PROCS   = 107
LABEL     "Blocked Processes"
;
```

En un archivo de especificación de clase se pueden incluir varias clases.

Al ejecutar el compilador DSI, **sdlcomp**, el modelo para registrar datos DSI se registra en el Almacén de datos de métricas.

Sintaxis de especificación de clase

Use las convenciones siguientes para crear un archivo de especificación de clase:

- Las instrucciones de sintaxis que se muestran entre corchetes `[ ]` son opcionales.
- 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.
- Los comentarios empiezan por # o //. Todo lo que siga a # o // en una misma línea será ignorado.
- Agregue un punto y coma después de cada descripción de clase y descripción de métrica.
- Las instrucciones no distinguen entre mayúscula y minúscula.

Nota: 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.

Por ejemplo:

```
CLASS class_name = class_id_number
[LABEL "class_label_name"]
[RECORDS PER HOUR  number]

;

METRICS
metric_name = metric_id_number
[ LABEL "metric_label_name" ]
;
```

Descripción de clase

Para crear una descripción de clase, asigne un nombre a un grupo de métricas desde un origen de datos determinado.

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` *nombre_de_clase* `=` *número_de_ID_de_clase*
`[LABEL`*"nombre_de_etiqueta_de_clase"]*

`[ RECORDS PER HOUR` *número*`]`

`;`

CLASS

El nombre de clase e ID de clase identifican un grupo de métrica de un origen de datos específico.

Sintaxis

`CLASS` *nombre_de_clase* `=` *número_de_ID_de_clase*

Instrucciones sobre su uso

nombre_de_clase y *número_de_ID_de_clase* deberán cumplir los requisitos siguientes:

- *class_name* es alfanumérico y puede tener un máximo de 20 caracteres. El *nombre_de_clase* puede empezar por un caracter alfabético y contener guiones bajos (si bien no caracteres especiales). No distingue entre mayúscula y

minúscula.

- *class_ID_number* es alfanumérico y puede tener un máximo de 6 caracteres.
- *class_name* y *class_ID_number* deben ser únicos entre las clases que se definan y no podrán coincidir con ninguna de las aplicaciones definidas en el archivo **parm** de Componente Performance Collection. Para más información sobre el archivo **parm**, consulte Uso del archivo parm.

Por 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 "nombre_de_etiqueta_de_clase" ]
```

Instrucciones sobre su uso

nombre_de_etiqueta_de_clase deberá cumplir los requisitos siguientes:

- Deberá estar entre comillas dobles.
- Deberá contener un máximo de hasta 48 caracteres.
- No puede coincidir con ninguno de los elementos de palabra clave de la sintaxis de especificación de clase DSI.
- 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 *nombre_de_clase*

Ejemplo

```
CLASS VMSTAT_STATS = 10001

LABEL "VMSTAT data";
```

Registros por hora

El ajuste `RECORDS PER HOUR` determina cuántos registros se escribirán en el archivo de base de datos cada 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 base de datos. El método usado para resumir los ítem de datos está especificado en la descripción de métrica. Para obtener más información, consulte Método de resumen.

Sintaxis

`[ RECORDS PER HOUR` *número*`]`

Cómo usar los registros por hora

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
```

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. . .

Por ejemplo

```
CLASS VMSTAT_STATS = 10001

LABEL  "VMSTAT data"

RECORDS PER HOUR  6;
```

En el siguiente ejemplo se escribirá un registro cada 10 minutos.

Configuración predeterminada

La configuración predeterminada de la descripción de clase será:

```
LABEL (class_name)
```

```
RECORDS PER HOUR 12
```

Para usar los valores predeterminados, introduzca sólo la palabra clave `CLASS` con un *nombre_de_clase* y un *número_de_ID_de_clase* numérico.

Especificación de clase de ejemplo

```
CLASS VMSTAT_STATS = 10001

LABEL "VMSTAT data"

RECORDS PER HOUR 120;

METRICS

RUN_Q_PROCS = 106

LABEL "Procs in run q"

;

BLOCKED_PROCS = 107

LABEL "Blocked Processes"

;

SWAPPED_PROCS = 108

LABEL "Swapped Processes"

;

AVG_VIRT_PAGES = 201

LABEL "Avg Virt Mem Pages"

;

FREE_LIST_SIZE = 202

LABEL "Mem Free List Size"

;

PAGE_RECLAIMS = 303

LABEL "Page Reclaims"

;

ADDR_TRANS_FAULTS = 304
```

```
LABEL "Addr Trans Faults"
;
PAGES_PAGED_IN = 305
LABEL "Pages Paged In"
;
PAGES_PAGED_OUT = 306
LABEL "Pages Paged Out"
;
PAGES_FREED = 307
LABEL "Pages Freed/Sec"
;
MEM_SHORTFALL = 308
LABEL "Exp Mem Shortfall"
;
CLOCKED_PAGES = 309
LABEL "Pages Scanned/Sec"
;
DEVICE_INTERRUPTS = 401
LABEL "Device Interrupts"
;
SYSTEM_CALLS = 402
LABEL "System Calls"
;
CONTEXT_SWITCHES = 403
LABEL "Context Switches/Sec"
;
USER_CPU = 501
```

```
LABEL "User CPU"

;

SYSTEM_CPU = 502

LABEL "System CPU"

;

IDLE_CPU = 503

LABEL "Idle CPU"

;
```

Descripciones de métrica

El nombre de métrica y el numero de ID identifican la métrica que se está recopilando.

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:

- El nombre de métrica puede contener un máximo de 20 caracteres.
- Debe comenzar por un carácter alfabético.
- El nombre de métrica solo podrá contener guiones bajos y caracteres alfanuméricos.
- No distingue entre mayúscula y minúscula.

El número ID de métrica puede contener un máximo de 6 caracteres.

Los parámetros *metric_name* y *metric_id_number* deben ser únicos entre 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 (;). 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.

Una métrica de marca de hora se inserta 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`.

Nota: 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.

ETIQUETA

La etiqueta de métrica identifica la métrica en datos exportados y gráficas de Performance Manager.

Sintaxis

`[LABEL "`*nombre_de_etiqueta_de_métrica*`"]`

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.

Nota: Si la etiqueta contiene dobles comillas, anteponga una barra diagonal inversa (\). Por ejemplo, debería introducir `"\"my\" data"` si la etiqueta fuese `"my" data`.

Por 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`. El método de resumen sólo es válido con métricas numéricas.

Sintaxis

`[{TOTALED | AVERAGED | SUMMARIZED BY` *nombre_de_métrica*`}]`

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`.

Por ejemplo:

```
METRICS
ITEM_1_3 = 11203
LABEL    "TOTAL_ORDERS"
TOTALED;
ITEM_1_5   = 11205
LABEL   "LINES_PER_ORDER"
SUMMARIZED BY ITEM_1_3;
```

Compilación del archivo de especificación de clase con el compilador DSI

Ejecute el compilador DSI, **sdlcomp** con el archivo de especificación de clase.

Sintaxis del compilador

```
sdlcomp <archivo_especificación_clase> <nombre_archivo_registro>
```

En este ejemplo:

`<archivo_especificación_clase>` es el nombre del archivo que contiene la especificación de clase. Si no está en el directorio actual, deberá aparecer el nombre completo.

`<nombre_archivo_registro>` se usa como nombre del origen de datos.

El compilador DSI **sdlcomp** no creará archivos basados en el nombre de archivo de registro. Por el contrario, se usa el nombre de archivo de registro como nombre del origen de datos.

Por ejemplo:

Si "/tmp/test_log" o "C:\test_log" se proporciona como el argumento establecido del archivo de registro (en la línea de comandos), se usará test_log como nombre del origen de datos.

El modelo para registrar datos DSI se guarda en el Almacén de datos de métricas.

Compilador sdlcomp

El compilador **sdlcomp** comprueba el archivo de especificación de clase por si se han producido errores. Si no es encuentran errores, agrega las descripciones del origen de datos, de clase y de métrica al Almacén de datos de métricas.

Cambio de especificación de clase

Para cambiar un archivo de especificación de clase, deberá recrear el conjunto de orígenes de datos como sigue:

1. Pare el proceso de `dsilog`.
2. Exporte los datos del Almacén de datos de métricas para todas las clases.
3. Ejecute `sdlutil` para eliminar el origen de datos. Consulte Gestión de datos con sdlutil para más información.
4. Actualice el archivo de especificación de clase.
5. Ejecute `sdlcomp` para volver a compilar la especificación de clase.
6. Ejecute `dsilog` para comenzar el proceso de archivo en base a la nueva especificación de clase.

Capítulo 16: Registro de métricas DSI en el Almacén de datos de métricas

Para registrar las métricas DSI en el Almacén de datos de métricas, asegúrese de que el modelo para registrar datos DSI se guardan en el almacén de datos. Modifique el archivo de definición de alarmas, **alarmdef**, para notificar cuándo las métricas DSI exceden las condiciones definidas. Para obtener más información sobre la definición de alarmas para métricas DSI, consulte Definición de alarmas para métricas DSI. Inicie el proceso de recopilación en la línea de comandos. Canalice los datos del proceso de recopilación al archivo **dsilog** (o use otro método que los lleve a **stdin**) con los conjuntos de opciones y las variables adecuadas.

Sintaxis

```
dsilog <sdl-file-name> <data-class-name> [options]
```

En este ejemplo:

`<sdl-file-name>` es el nombre del archivo autodescriptivo raíz.

`<data-class-name>` es el nombre de la clase de datos.

Tabla 1: Parámetros y opciones de `dsilog`

Variables y opciones	Definiciones
-c <char>	Usa el caracter especificado como separador/delimitador de cadena.
-i <fifo>	Indica que la entrada debe venir del `fifo` nombrado. El fifo es la alternativa al stream de entrada.
-timestamp	La marca de hora que debe especificarse como la primera columna en el archivo de datos de entrada. Tiene que estar en formato EPOCH.
-?	Muestra la descripción de sintaxis.
-vers	Muestra la información de versión de impresión.

Nota: 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 Ejemplos de integración de fuentes de datos, donde encontrará ejemplos de scripts problemáticos y recomendados.

Proceso de registro dsilog

El proceso `dsilog` requiere que el usuario cree su propio programa o que use un proceso existente. 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 name> <class name>
```

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 se ejecute en segundo plano.

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 el delimitador predeterminado.

Deberá incluirse un caracter de línea nuevo al final del registro ingresado en DSI con objeto de que DSI lo interprete correctamente.

Administrar datos con sdlutil

Use el programa `sdlutil` para gestionar los datos de los archivos de base de datos DSI. Puede hacer lo siguiente:

- Enunciar la información de métrica y la clase definida en `stdout`. Es posible redireccionar la salida a un archivo.
- Suprimir el origen de datos, clases, métricas y datos del Almacén de datos de métricas.
- Mostrar información de la versión.

Sintaxis

`sdlutil <nombre_de_archivo_de_registro>[`*opción*`]`

Variables y opciones	Definiciones
`logfile name`	se usa como nombre del origen de datos.
`-rm all`	suprime el origen de datos, clases, métricas y datos del Almacén de datos de métricas.
`-vers`	muestra la información de versión.
`-?`	muestra la descripción de sintaxis.

Por ejemplo:

Ejecuta el comando siguiente para suprimir el origen de datos, `test_log` junto con las clases, métricas y datos del Almacén de datos de métricas.

```
sdlutil /tmp/test_log -rm all
```

Capítulo 17: Uso de los datos DSI registros en el Almacén de datos de métricas

Después de que los datos DSI se registra en el Almacén de datos de métricas, puede exportar los datos con el programa **extract** de Componente Performance Collection. Asimismo, 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.

Definición de alarmas en métrica DSI

Use el archivo `alarmdef` en el sistema de Componente Performance Collection para definir alarmas en las métricas DSI. Estas alarmas notifican al usuario si la métrica DSI excede las condiciones establecidas. 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 *nombre_de origen_datos* y el *nombre_clase* como se muestra a continuación:

```
nombre_origen_datos:nombre_clase:nombre_métrica
```

- *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 se usará. Para obtener más información, consulte la instrucción USE en el capítulo *Alarmas de rendimiento*.

Para activar los cambios realizados en el archivo `alarmdef`, ejecute el comando `ovpa restart alarm` en la línea de comandos.

Para obtener información detallada sobre la sintaxis de definición de alarmas, cómo se procesan las alarmas y cómo se personalizan las definiciones de alarmas, consulte Alarmas de rendimiento.

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 acciones locales pueden ser realizadas en el sistema de Componente Performance Collection. La información de alarma también se puede ser enviar a HP Operations Manager.

Exportación de datos DSI

Para exportar los datos de DSI desde el Almacén de datos de métricas, use la función `export` del programa `extract`. Para obtener más información, consulte, Uso del programa Extract. Use el comando `ovcodautil -obj` para ver la lista de clases y orígenes de datos de DSI que se pueden exportar.

Ejemplo de uso del programa extract para exportar datos del archivo de registro DSI

`extract -xp -l logfile_set -C class [`*opciones*`]`

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.
- 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

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.

Capítulo 18: Ejemplos de integración de orígenes de datos (DSI)

La integración de orígenes de datos es una tecnología potente y flexible. Este capítulo contiene ejemplos sobre cómo usar DSI para recopilar y registrar datos en el Almacén de datos de métricas.

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.

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 que se usa 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 después especificar una ruta de archivo de registro. El nombre del archivo de registro en la ruta de archivo de registro se usa como nombre del origen de datos.

Nota: No se ha creado ningún archivo de registro.

En el ejemplo siguiente de una salida de compilador y de comandos, se utiliza **/tmp/vmstat.spec** como nombre de archivo de especificación de clase y /tmp/VMSTAT_DATA como ruta de archivo de registro.

```
-> sdlcomp /tmp/vmstat.spec /tmp/VMSTAT_DATA


This example creates a datasource called VMSTAT_DATA. If there are syntax
```

```
errors in the class specification file, messages indicating the problems
are displayed and the datasource is not created.
```

Para comprobar que la clase VMSTAT_STATS se ha agregado correctamente al origen de datos, ejecute el comando siguiente:

```
ovcodautil -ds VMSTAT_DATA -obj
```

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:

```
La métrica tiene datos no válidos. Ignorar hasta el final de línea, el valor
de métrica excede el máximo.
```

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

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
```

Hay que tener en cuenta que para exportar datos DSI, hay que ser el usuario raíz o el creador del archivo de registro.

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"
;
METRICS
who_wc = 151
label "who wc"
averaged
;
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
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 &
   ```

Capítulo 19: Uso de la transmisión de métricas

HPE Operations Agent permite registrar las métricas personalizadas en el almacén de datos de métricas, junto con las clases de métricas de rendimiento del sistema predeterminadas.

Con HPE Operations Agent 12.01, las métricas de rendimiento personalizadas y del sistema también están disponibles para su transmisión mediante la funcionalidad Transmisión de métricas. Puede configurar la transmisión de métricas mediante la directiva **Configuración de la transmisión de métricas**.

La transmisión de métricas permite transmitir datos de las métricas a un suscriptor de destino (por ejemplo: el motor de rendimiento) y usar los datos para gráficos y análisis. Puede enviar datos a **hpsensor** e implementar la directiva **Configuración de la transmisión de métricas** desde HPE Operations Manager i (OMi) a HPE Operations Agent para transmitir datos de las métricas. Para obtener más información acerca de la directiva **Configuración de la transmisión de métricas**, consulte la *Guía de administración de OMi*.

Nota: HPE Compute Sensor (hpsensor) es un proceso liviano de recopilación de datos de **rendimiento** y **registro**.

Transmisión de métricas

Para transmitir datos de métricas, a interfaz basada en REST se proporciona para enviar datos a **hpsensor**. **hpsensor** publicará los datos enviados al suscriptor de destino con el intervalo configurado.

Antes de enviar los datos a **hpsensor**, debe publicar la información de registro en **hpsensor** mediante la API de REST proporcionada. Además es necesario crear una directiva en HPE Operations Manager i (OMi), en el tipo de directiva **Configuración de la transmisión de métricas** para elegir la lista de métricas que se van a transmitir, proporcionar la URL de destino e implementar dicha directiva en HPE Operations Agent desde OMi.

Nota: Las métricas no se transmiten si el receptor de destino no está disponible.

Las siguientes API de REST están disponibles para su registro y el envío de datos desde el sistema local:

1. Registro: **`/v4/RTCM/Register`**
2. Envío de datos: **`/v4/RTCM/SubmitData`**

Uso:

Se puede utilizar el siguiente URI para publicar datos en **hpsensor** con un cliente http:

- Para el registro: **`http://<hostlocal>:<puerto>/hpcs/v4/RTCM/Register`**
- Para el envío de datos: **`http://<hostlocal>:<puerto>/hpcs/v4/RTCM/SubmitData`**

En este ejemplo:

`<hostlocal>` es un nombre de host que resuelve la dirección del bucle de retroceso. La dirección del bucle de retroceso es: `127.0.0.1` para IPv4 y `[::1]` para IPv6.

`<puerto>` es el puerto BBC (de forma predeterminada, 383) o el puerto **hpsensor**.

hpsensor acepta datos en formato **v4 JSON** (JavaScript Object Notation). La importancia del formato v4 JSON es el registro de datos centrados en las métricas y el envío de datos. Los datos deben contener los siguientes objetos:

1. Título
2. Versión
3. Metric Payload, formado por los campos MetaData y Data

A continuación se proporciona el esquema en formato v4 JSON:

```
[{
"Title": "Real Time Metrics",
"Version": "v4",
"Metric Payload": [
     {
     "MetaData": {
          "MetricName": {
```

```
                "description": "The unique name of the
metric",
                "type": "string"
           },
           "CategoryType": {
                "description": "The category type of the
metric (attribute/counter/gauge/...)",
                "type": "string"
           },
           "DataType": {
                "description": "The data type of the metric",
                "type": "string"
           },
           "ClassName": {
                "description": "The class name of the metric",
                "type": "string"
           },
           "Unit": {
                "description": "The unit of the metric value",
                "type": "string"
           },
           "Label": {
                "description": "The label of the metric",
                "type": "string"
           },
           "DataSource": {
                "description": "The data source the metric
belongs to",
```

```
                "type": "string"
            },
            "Key": {
                "description": "Key number of the metric, key
value 0... n for Key metric",
                "type": "integer"
            },
            "Interface": {
                "description": "The interface of collection,
either Legacy or Stream",
                "type": "string"
            },
            "CollectorName": {
                "description": "The name of the metric
collector",
                "type": "string"
            },
            "CollectorExec": {
                "description": "The collector executable to be
executed",
                "type": "string"
            },
            "OriginalMetricName": {
                "description": "The domain the metric belongs
to",
                "type": "string"
            },
            "required": [
                "MetricName",
```

```
                "ClassName",
                "DataSource",
                "Key"
                     ]
          },
          "Data": {
                "Instances": [
                     {
                          "value": {
                               "description": "The value of
the metric for an instance",
                               "type": "string/integer/real"
                          },
                          "timestamp": {
                               "description": "The epoch time
when the metric was collected",
                               "type": "integer"
                          },
                          "dimensions": {
                               "keyinstance": {
                                    "description": "The key
to uniquely identify an instance",
                                    "type":
"string/integer/real"
                               },
                               "minItems": 1,
                               "uniqueItems": true
                          },
                          "minItems": 1,
```

```
                        "uniqueItems": true
                    }
                ]
            }
        }
]
}]
```

Los datos están compuestos por los siguientes parámetros:

Parámetro	Obligatorio		Description
	Para el registro	Para el envío de datos	
Title	Sí	Yes	Debe ser **"Real Time Metrics"**.
Versión	Sí	Yes	Debe ser **"v4"**, ya que la API de REST proporcionada afecta datos en formato v4 JSON.
MetricName	Sí	Sí	El nombre exclusivo de la métrica.
CategoryType	*Opcional*	*Opcional*	El tipo de categoría de la métrica. Los tipos de categoría aceptados son: Key, Attribute, Gauge y Counter.
DataType	*Opcional*	*Opcional*	El tipo de datos de la métrica. Los tipos de datos aceptados son: INT32, UINT32, DOUBLE, INT64, UINT64, STRING, BOOL y TIME.
Description	*Opcional*	*Opcional*	Descripción de la métrica.
Label	*Opcional*	*Opcional*	Etiqueta de la métrica.
Unit	*Opcional*	*Opcional*	Unidad de la métrica. Por ejemplo: %, kbps, mbps, etc.

Key	Sí	Sí	Este campo identifica si esta métrica es de claves. El valor **-1** significa que no es una clave y el valor 0 o superior significa que es una métrica de claves. El valor de la métrica de clave define las distancias de la clase.
ClassName	Sí	Yes	El nombre de clase de la métrica.
DataSource	Sí	Sí	El origen de datos de la métrica.
CollectorName	Sí	*Opcional*	El nombre del recopilador de la métrica.
CollectorExec	*Opcional*	*Opcional*	La ruta al recopilador de la métrica con el comando ejecutable.
Interface	Sí	*Opcional*	La interfaz de la métrica. Stream o Legacy. Stream se utiliza cuando se desea transmitir métricas a **hpsensor** sin registrarla en el almacén de datos de métricas y Legacy se utiliza cuando se desea registrar los datos en el almacén de datos de métricas usando interfaces heredadas, como DDF, o el envío masivo y también transmitirlos a **hpsensor**. Si no se especifica, la interfaz predeterminada es Legacy.
value	*Must para la métrica de claves*	Sí	Valor de la métrica. El valor especificado debe coincidir con el DataType de la métrica.
timestamp	*Must para la métrica de claves*	Sí	El tiempo epoch cuando se recopila el valor de la métrica.
dimensions	*Opcional*	Sí	El nombre de la instancia de la clave de la métrica.

Registro

La API de REST `/v4/RTCM/Register` se utiliza para publicar la información de registro en **hpsensor**. La información de registro debe contener los siguientes objetos:

1. Título
2. Versión
3. Metric Payload, formado por los campos MetaData y Data (para las instancias de métrica **Key**)

Por ejemplo: Suponga un caso de uso en el que desea publicar la información de registro en **hpsensor** para transmitir datos para 2 métricas personalizadas (1 métrica y 1 métrica de clave). Para este escenario, se puede publicar la siguiente información de registro en formato v4 JSON:

```
[{
"Title": "Real Time Metrics",
"Version": "v4",
"Metric Payload": [
     {
     "MetaData": {
          "MetricName": "Metric_ACTIVE_PROC",
          "CategoryType": "Attribute",
          "DataType": "DOUBLE",
          "Description": "Desc",
          "Label": "Active Proc",
          "Unit": "N/A",
          "Key": -1,
          "ClassName": "Class",
          "DataSource": "DataSouce",
          "CollectorName": "Collector",
```

```
          "CollectorExec":
"$OvDataDir$bin/instrumentation/RTCM_collector.sh Collector",
          "Interface": "Stream"
     }
},
     {
     "MetaData": {
          "MetricName": "Instance_NAME",
          "CategoryType": "Attribute",
          "DataType": "STRING",
          "Description": "Desc",
          "Label": "Metric Name1",
          "Unit": "N/A",
          "Key": 0,
          "ClassName": "Class",
          "DataSource": "DataSOurce",
          "CollectorName": "Collector",
          "CollectorExec":
"$OvDataDir$bin/instrumentation/RTCM_collector.sh",
          "Interface": "Stream"
     },
     "Data": {
          "Instances": [{
               "value": "inst_1",
               "timestamp": 1459724247
          }, {
               "value": "inst_2",
               "timestamp": 1459724247
```

```
            }]
      }
      }
      ]
}]
```

Utilice la API de REST **`/v4/RTCM/Register`** para publicar la información de registro en **hpsensor**. Para obtener más información sobre el uso de esta API de REST, consulte la sección de uso.

Envío de datos

La API de REST **`/v4/RTCM/SubmitData`** se utiliza para enviar datos de métrica personalizada a **hpsensor**. Los datos enviados deben contener los siguientes objetos:

1. Título
2. Versión
3. Metric Payload, formado por los campos MetaData y Data (para todas las métricas)

Nota: Para el envío de datos, es necesario especificar las instancias usando la etiqueta de dimensiones para todas las métricas; de lo contrario, de descartará el envío de datos.

Por ejemplo: Suponga un caso de uso en el que desea enviar los datos de la métrica personalizada a **hpsensor** para transmitir datos para 2 métricas personalizadas (1 métrica y 1 métrica de clave). En este caso, se pueden enviar los siguientes datos en formato v4 JSON:

```
[{
"Title": "Real Time Metrics",
"Version": "v4",
"Metric Payload": [
      {
      "MetaData": {
```

```
          "MetricName": "Metric_ACTIVE_PROC",
          "CategoryType": "Attribute",
          "DataType": "DOUBLE",
          "Description": "Desc",
          "Label": "Active Proc",
          "Unit": "N/A",
          "Key": -1,
          "ClassName": "Class",
          "DataSource": "DataSource"
     },
     "Data": {
          "Instances": [{
               "value": "0.0",
               "timestamp": 1459725391,
               "dimensions": {
                    "Instance_NAME": "inst_1"
                         }
          }, {
               "value": "0.0",
               "timestamp": 1459725391,
               "dimensions": {
                    "Instance_NAME": "inst_2"
                         }
          }]
     }
},
     {
```

```
"MetaData": {
     "MetricName": "Instance_NAME",
     "CategoryType": "Attribute",
     "DataType": "STRING",
     "Description": "Desc",
     "Label": "Metric Name1",
     "Unit": "N/A",
     "Key": 0,
     "ClassName": "Class",
     "DataSource": "DataSource"
},
"Data": {
     "Instances": [{
          "value": "inst_1",
          "timestamp": 1459725391,
          "dimensions": {
               "Instance_NAME": "inst_1"
                    }
     },{
          "value": "inst_2",
          "timestamp": 1459725391,
          "dimensions": {
               "Instance_NAME": "inst_2",
                    }
     }]
}
]
```

```
}]
```

Utilice la API de REST **`/v4/RTCM/SubmitData`** para enviar los datos de la métrica a **hpsensor**. Para obtener más información sobre el uso de esta API de REST, consulte la sección de uso.

Caracteres especiales en el registro y el envío de datos

hpsensor acepta solicitudes de datos en formato de URI y es posible que haya caracteres especiales en MetricName, DataSource, ClassNames e Instance_ NAME. Los caracteres especiales admitidos y no admitidos para diversas entidades son los siguientes:

Entity	Caracteres especiales admitidos	Caracteres especiales no admitidos
MetricName/DataSource/ClassName	_~`!@$%^&*()-+= {[]}|_;<>.?'	/,#\:”
InstanceName	_~`!@$%^&*-+={} |_.?:'	/,#\”[]();<>

Nota: Para el registro y el envío de datos:

- El número de campos MetaData puede variar.
- El número distancias de una métrica puede variar, y no todas las métricas necesitan tener el mismo número de instancias.
- Puede haber una o varias dimensiones para una estancia.
- Se puede utilizar la combinación de caracteres `<*>` como un comodín en la directiva **Configuración de la transmisión de métricas**. Por ejemplo: `"Instances": ["(example)","(<*>.company.com)"]`
- Se pueden utilizar los caracteres no admitidos mencionados para InstanceName, pero no se pueden usar en la directiva **Configuración de la transmisión de métricas** para la coincidencia exacta de dichos caracteres. En su lugar, puede usar un comodín `<*>` para que coincida cualquier carácter.

Configuración del intervalo de publicación

hpsensor los datos de las métricas de rendimiento personalizadas y del sistema en el suscriptor de destino con el intervalo configurado. De forma predeterminada, el intervalo de publicación es de 10 segundos. Puede usar la siguiente variable XPL para configurar el intervalo de publicación predeterminado:

Variable	Espacio de nombres	Descripción	¿Reinicio necesario?	Valor predeterminado	Tipo
PUBLISH_INTERVAL	`hpsensor`	Este parámetro establece el intervalo en el que los datos de las métricas de rendimiento personalizadas y del sistema se publican en el suscriptor de destino.	No	10	Entero

Para configurar el intervalo de publicación predeterminado, siga estos pasos:

1. Inicie una sesión en el nodo HPE Operations Agent como administrador.
2. Ejecute el comando siguiente:

   ```
   <OvInstallBinDir>ovconfchg -ns hpsensor -set PUBLISH_INTERVAL <valor>
   ```

El valor predeterminado es 10 (en segundos) y el valor mínimo permitido es 5 (en segundos).

Nota: El intervalo de publicación establecido será aplicable a todos los destinos.

Utilización de recursos y escalabilidad

Los datos enviados a **hpsensor** para la transmisión de métricas se almacenarán en memoria, y **hpsensor** solo mantiene un valor de instancia de una métrica. Si envía más puntos de datos, habrá una gran utilización de la memoria. Hewlett Packard Enterprise recomienda que el número de puntos de datos enviado no sea superior a 100 000 para una utilización de la memoria óptima.

Parte V: Seguimiento de transacciones

HPE Operations Agent puede rastrear las transacciones cuando procesan por las aplicaciones. El 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).

Las transacciones se monitorizan solo después de que el registro de transacciones se active en el archivo **parm**. 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 el componente Performance Collection o GlancePlus pueden usarlo.

El rastreo de transacciones proporciona una vista de cliente del tiempo transcurrido desde el comienzo al final de una transacción, le ayuda a gestionar los acuerdos de nivel de servicio (SLA) y a generar alarmas cuando los objetivos de nivel de servicio (SLO) exceden las condiciones definidas en el archivo `alarmdef`.

Capítulo 20: 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 HPE Operations Agent y HPE 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 transacción

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 24, 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 25, 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 de la duración necesaria 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 25, 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*.

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 buscapersonas de un técnico, de forma que los problemas puedan investigarse y resolverse inmediatamente. Para más información, consulte el capítulo 26, 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 Lista de clases para una fuente de datos y, a continuación, **graph transaction metrics** para varias transacciones. Para obtener más información, consulte la *ayuda en línea de Performance Manage*r.

... 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 recopilador **oacore** 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 el capítulo 21)
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 el capítulo 21)
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 el capítulo 21).
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 el capítulo 26)
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 del capítulo 22).
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 el capítulo 21).
7. Limite el número de nombres de transacción exclusivos que se instrumentan. (Consulte Limites en transacciones únicas en el capítulo 22)
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 de una llamada APi deARM en el capítulo 21)
9. Use bibliotecas compartidas para la vinculación. (Consulte la sección "Ejemplos de opción de compilador C por plataforma" en la página 381).

Capítulo 21: 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 archivo de configuración de transacciones, `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/cmgarmw

Para obtener información acerca del uso del agente de registro, consulte *Application Response Measurement 2.0 API Guide*.

Nota: 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.

`arm_init()`	Asigna un nombre al usuario y registra la aplicación y (opcionalmente) el usuario.
`arm_getid()`	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.
`arm_start()`	Señala el inicio de un caso de transacción exclusiva.
`arm_update()`	Actualiza los valores de un caso de transacción exclusiva.
`arm_stop()`	Señala el final de un caso de transacción exclusiva.
`arm_end()`	Señala el final de la aplicación.

Consulte el manual Application Response Measurement 2.0 API Guide 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, 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
```

Nota: 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` comienza cuando se inicia el recopilador de datos **oacore** de Componente Performance Collection con el comando `ovpa start`. `ttd` se ejecuta en modo de segundo plano cuando se envía 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).

Precaución: 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 el script `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 `/<InstallDir>/bin/ttd -k`. Sin embargo, no recomendamos detener `ttd`, excepto al reinstalar Componente Performance Collection.

Si Componente Performance Collection no está en el 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 `EPIPEerrno` en las 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_instalación/bin/ovpa start` o `/<dir_instalación/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 `ttd`) 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 `ttd` 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 `ttd` 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,

`ttd` 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`ttd`.

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 el proceso `perfd` de **oacore** o cuando se inicia GlancePlus.

Componente Performance Collection y GlancePlus requieren que midaemon y `ttd` 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 `mideamon`, en el orden correcto. GlancePlus inicia `mideamon`, si aún no se está ejecutando. `midaemon` inicia `ttd`, si aún no se está ejecutando.

Consulte la sección "Sobrecargade la CPU " en la página 352 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, en 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 nuevasaplicaciones

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 **oacore** 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 **oacore** 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
`tran`	`tran=`*nombre_transacción*	Obligatorio
`slo`	`slo=`*seg.*	Opcional
`range`	`range=`*seg. [,seg.,...]*	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 **oacore**, 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 **oacore** 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 la página 341 de 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

```
[MANufactr]

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 `MANufctr`, `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 las correlaciones de ARM, consulte la sección de temas avanzados en *Application Response Measurement 2.0 API Guide*. 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 deminio del recopilador de Componente Performance Collection, **oacore** genera archivos de base de datos en el disco.

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 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 **oacore** será un poco mayor en un sistema con aplicaciones instrumentadas con llamadas a ARM 2.0. Solo las aplicaciones que están instrumentadas con llamadas a ARM 2.0 y usan mucho las correlaciones o métricas definidas por el usuario tendrán un importante impacto en el rendimiento en `midaemon`, **oacore** 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 la guía *Application Response Measurement 2.0 API Guide*, 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*).

Capítulo 22: 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 "Funcionamiento del seguimiento de transacciones" en la página 338. Consulte el "Ejemplos de seguimiento de transacciones" en la página 364 para ver ejemplos.

Antes de comenzar

Componente Performance Collection proporciona la biblioteca compartida `libarm.*` en las siguientes ubicaciones:

Plataforma	Ruta
IBM RS/6000	`/usr/lpp/perf/lib/`
Otras plataformas UNIX	`/opt/perf/lib/`

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 381, al final de este manual. Consulte también la sección sobre la biblioteca compartida de ARM (`libarm`) en la guía *Application Response Measurement 2.0 API Guide* para ver información acerca de cómo obtenerla. Para ver una descripción de `libarm`, consulte "Biblioteca ARM (libarm)" en la página 374, 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 `parm` de Componente Performance Collection 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*.
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 la fuente de datos y vuelva a conectarse a ella.
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 **oacore** mediante `ovpa stop`.
   c. Detenga el uso de Glance.
   d. Detenga `ttd` con `ttd -k`.

   Cuando haya realizado los cambios:
   a. Reinicie **oacore** 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 `parm` de Componente Performance Collection 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 activar el archivo `parm` actualizado como se describe a continuación:

Estado de Componente Performance Collection	Comando para activar el seguimiento de transacciones
En ejecución	`ovpa restart`
No ejecutándose	`ovpa start`

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 359 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. 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 "Funcionamiento del seguimiento de transacciones" en la página 338 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 recopilador **oacore** mediante `ovpa stop`.
3. Detenga el uso de Glance.
4. Detenga `ttd` con `ttd -k`.

Cuando haya realizado los cambios:

1. Reinicie **oacore** 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 Lista de clases de una fuente de datos en Performance Manager y, a continuación, representar en gráfico las métricas de varias transacciones. Para obtener más información, consulte la ayuda en línea de Performance Manager, a la que puede acceder mediante el menú Ayuda de Performance Manager. Si no ve las transacciones que esperaba en Performance Manager, cierre la fuente de 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.

Capítulo 23: 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 nº 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.

Capítulo 24: 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 `/<dir_instalación>/paperdocs/gp/C/` como `gp-metrics.txt`.

  `dir_instalación` 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 *Dictionary of Operating System Performance Metrics de HPE Operations* de la plataforma en las ubicaciones siguientes:

  En UNIX/Linux, en `/<dir_instalación>/paperdocs/ovpa/C/` como `met<`*plataforma*`>.txt`.

  En Windows en *%ovinstalldir%*`paperdocs\ovpa\C` como `met<`*plataforma*`>.txt.`

Capítulo 25: 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 "Descripción del seguimiento de transacciones" en la página 332

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 "Descripción del seguimiento de transacciones" en la página 332. 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 "Funcionamiento del seguimiento de transacciones" en la página 338

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
```

Capítulo 26: 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 **oacore**

Cómo se usan los tipos de datos

En la tabla siguiente se describe cómo se usan los tipos de datos en Componente Performance Collection. Es un complemento de "Data Type Definitions" en la sección "Advanced Topics"·de la guía *Application Response Measurement 2.0 API Guide*.

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 obtener 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 "Advanced Topics"·de la guía *Application Response Measurement 2.0 API Guide*. 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

Tabla 4: Ejemplos de lo que se registra con tipos de datos específicos del programa, continuación

…lo que pasa el programa	…lo que se registra
EJEMPLO 2 String32 NumericID32 NumericID64	 NumericID32 NumericID64
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.

Capítulo 27: Bibliotecas de transacciones

Este apéndice incluye los componentes siguientes:

- Biblioteca Application Response Measurement (`libarm`)
- Ejemplos de opción de compilador C por plataforma
- 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

`/opt/perf/lib/`	`libarm.0`	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 `-larm`, el cargador 11.0 referenciará automáticamente esta biblioteca.
	`libarm.1`	Biblioteca compartida compatible HP-UX 11 (segura para subprocesos). Esta

		biblioteca será referenciada por programas que hayan sido vinculados con `-larm` o versiones HP-UX. Si un programa vinculado con 10.20 referencia esta biblioteca, (por ejemplo, si no ha sido vinculado con `-L /opt/perf/lib`, puede que aborte con un error del tipo "`/usr/lib/dld.sl: Unresolved symbol: _thread_once (code) from libtt.sl`".
	`libarm.sl`	Vínculo simbólico a `libarm.1`
	`libarmNOP.sl`	Biblioteca compartida “sin-operaciones” para ARM (las llamadas API ocurren pero no hacen nada; usada para pruebas en sistemas que no tienen instalado Componente Performance Collection.
`/opt/perf/examples/arm`	`libarmjava.sl`	Biblioteca compartida de 32 bits para ARM.
`/opt/perf/examples/arm/arm64`	`libarmjava.sl`	Biblioteca compartida de 64 bits para ARM.
`/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 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 342 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 64 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/arm64`	`libarmjava.so`	Biblioteca compartida

		de 32 bits para ARM.
`/opt/perf/examples/arm/arm64`	`libarmjava.so`	Biblioteca compartida de 64 bits para ARM.

Nota: 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.
`/opt/perf/lib/`	`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.

Nota: 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
  ```

Nota: 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, `/<dir_instalación>/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/`. `dir_instalación` es el directorio en el que se ha instalado Componente Performance Collection.

Ejemplos

Existen ejemplos de contenedores Java en el directorio `/<dir_instalación>/examples/arm/` directory. 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" en la página 385) 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 *Application Response Measurement 2.0 API Guide*.

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:

`ARMConstants.ARM_GOOD.`	Use este valor cuando la operación transcurra con normalidad y según lo esperado.	`ARMConstants.ARM_GOOD.`
`ARMConstants.ARM_ABORT.`	Use este valor cuando se produzca un fallo grave en el sistema.	`ARMConstants.ARM_ABORT.`
`ARMConstants.ARM_FAILED.`	Use este valor en aplicaciones en las que la transacción transcurra con normalidad, pero no se generen resultados.	`ARMConstants.ARM_FAILED.`

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 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`	• `ARMConstants. ARM_GOOD` Use este valor cuando la operación transcurra con normalidad y según lo esperado. • `ARMConstants.ARM_ABORT` Este valor se usa cuando se produce un error 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 `/<dir_instalación>/examples/arm/`, la cual incluye documentación html para cada clase Java. Comience con `index.htm`.

Parte VI: 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 HPE Operations Agent.

Componente Operations Monitoring

- *Problema:* Si HPOM gestiona un gran número de nodos (más de 1024), puede sufrir problemas de comunicación entre HPOM y los nodos gestionados. Es problema también puede aparecer cuando HPE Operations Agent se instala en el servidor de HP Performance Manager que se comunica con un gran número de nodos gestionados (más de 1024).

  Solución:

  Para evitar este problema, vaya al servidor de gestión (si HPOM gestiona más de 1024 nodos) o al servidor de HP Performance Manager (si el agente está instalado en un servidor de HP Performance Manager que se comunica con más de 1024 nodos) y siga estos pasos de configuración:

  Inicie sesión como raíz o administrador.

  Ejecute el comando siguiente:

  En las versiones de Windows de 32 bits:

  ```
  %ovinstalldir%bin\ovconfchg -ns xpl.net -set SocketPoll true
  ```

  En las versiones de Windows de 64 bits:

  ```
  %ovinstalldir%bin\win64\ovconfchg -ns xpl.net -set SocketPoll
  true
  ```

  En HP-UX/Linux/Solaris:

  ```
  /opt/OV/bin/ovconfchg -ns xpl.net -set SocketPoll true
  ```

  Reinicie el agente:

  En las versiones de Windows de 32 bits:

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

  En las versiones de Windows de 64 bits:

  ```
  %ovinstalldir%bin\win64\opcagt -stop
  %ovinstalldir%bin\win64\opcagt -start
  ```

  En HP-UX, Linux o Solaris:

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

- *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, los scripts Perl no funcionan en las directivas.

  Causa: Los scripts Perl disponibles en las directivas requieren la variable de configuración `PATH` para incluir el directorio en el que Perl (suministrado con HPE 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\a\bin"
     ```

  b. Reinicie el agente ejecutando los siguientes comandos:

     i. `ovc -kill`

     ii. `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 2:

  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 3: Abra el perfil o archivo de trabajo del nodo y asegúrese de que no incluyan configuraciones conflictivas para las variables.

- *Problema:* Cuando se despliega una directiva de interceptor de mensajes (msgi) particular y se desencadena un mensaje que coincide con la condición de la directiva, el proceso `opcmsgi` deja de responder con una elevada utilización de la CPU (consume el 25 por ciento en un servidor con 4 CPU, por tanto un 100 por ciento de una única CPU) y deja de procesar mensajes.

  Causa:

  El proceso `opcmsgi` deja de responder y ya no proporciona más mensajes debido al patrón siguiente:

  ```
  <*><@.variable><*>
  ```

  Solución:

  Sustituya el patrón `<*><@.variable><*>` por `<*><@.variable><S><*>`.

  Por ejemplo:

  Patrón original:

  ```
  ^\tM_Type: <@.SiSMonType><*>SiS: <
  [http://<@.SiSIP>:<@>].SiSURL><*>Monitor:
  <@.SiSMonName><*>Group: <@.SiSGrpName><*>Status:
  <@.SiSMonStatus><*>URL: <@.SiSMonURL><*>Match Content:
  <@.SiSMonMatch><*>Monitor Description: DOTCOM_BACKEND:
  {<@.OMTargetNode>}{<@.OMMsgGrp>}{<*.OMAppl>}{<*.OMObj>}
  {<*.OMSvcID>}{<@.SiSEMSCode>}{<*.OMCIs>}{<*.OMMsgText>}
  ```

  Patrón sustituido:

  ```
  ^\tM_Type: <@.SiSMonType><S><*>SiS: <
  [http://<@.SiSIP>:<@>].SiSURL><S><*>Monitor:
  <@.SiSMonName><S><*>Group: <@.SiSGrpName><S><*>Status:
  <@.SiSMonStatus><S><*>URL: <@.SiSMonURL><S><*>Match Content:
  <@.SiSMonMatch><S><*>Monitor Description: DOTCOM_BACKEND:
  {<@.OMTargetNode>}{<@.OMMsgGrp>}{<*.OMAppl>}{<*.OMObj>}
  {<*.OMSvcID>}{<@.SiSEMSCode>}{<*.OMCIs>}{<*.OMMsgText>}
  ```

  Por ejemplo:

  Patrón original:

```
^\tM_Type: <@.SiSMonType><*>SiS: <
[http://<@.SiSIP>:<@>].SiSURL><*>Monitor:
<@.SiSMonName><*>Group: <@.SiSGrpName><*>Status:
<@.SiSMonStatus><*>URL: <@.SiSMonURL><*>Match Content:
<@.SiSMonMatch><*>Monitor Description: DOTCOM_AUTONOMY:
{<@.OMTargetNode>}{<@.OMMsgGrp>}{<*.OMAppl>}{<*.OMObj>}
{<*.OMSvcID>}{<@.SiSEMSCode>}{<*.OMCIs>}{<*.OMMsgText>}
```

Patrón sustituido:

```
^\tM_Type: <@.SiSMonType><S><*>SiS: <
[http://<@.SiSIP>:<@>].SiSURL><S><*>Monitor:
<@.SiSMonName><S><*>Group: <@.SiSGrpName><S><*>Status:
<@.SiSMonStatus><S><*>URL: <@.SiSMonURL><S><*>Match Content:
<@.SiSMonMatch><S><*>Monitor Description: DOTCOM_AUTONOMY:
{<@.OMTargetNode>}{<@.OMMsgGrp>}{<*.OMAppl>}{<*.OMObj>}
{<*.OMSvcID>}{<@.SiSEMSCode>}{<*.OMCIs>}{<*.OMMsgText>}
```

- *Problema:* En HPE Operations Agent, el patrón coincidente informa de resultados incorrectos.

  Causa:

  En HPE Operations Agent, el patrón coincidente puede informar de resultados incorrectos si hay más de 148 operadores OR (|) en el patrón.

  > **Nota:** En la coincidencia de patrones, el operador OR (|) es un carácter especial que separa dos expresiones haciendo coincidir la cadena en ambas expresiones.
  >
  > **Por ejemplo:**
  >
  > `[ab|c]d`
  >
  > En el patrón anterior, el operador OR <|> coincide con la cadena **abd** y la cadena **cd**.

  Solución:

  Asegúrese de que el número de operadores OR (|) no excede de 148.

- *Problema:* El proceso `opcmona` se reinicia automáticamente después de que se ejecuta una directiva de tareas programadas con un script Perl incrustado en el nodo y aparece el siguiente mensaje en la consola HPOM:

  ```
  (ctrl-208) El componente 'opcmona' con el pid 6976 salió con el valor
  de salida '-1073741819'. Reiniciando el componente.
  ```

  Causa:

Referencias de `exit (o)` en el script Perl incrustado causan que `opcmona` se reinicie.

Solución:

No use `exit (o)` en el script Perl incrustado.

- *Problema:* En el sistema Red Hat Enterprise Linux, ovbbccb no puede acceder al archivo de zona horaria y escribe la información de fecha en formato UTC (GMT) en lugar de en la zona horaria actual establecida por el sistema. Esta información de zona horaria errónea se registra en el archivo `System.txt`.

  Solución:

  Para establecer la zona horaria correcta, siga estos pasos:

  a. Ejecute el comando siguiente para comprobar la ubicación del archivo zoneinfo:

     ```
     ls /usr/share/zoneinfo/
     ```

  b. Ejecute el comando siguiente para comprobar la ubicación del archivo localtime:

     ```
     ls -l /etc/localtime
     ```

     Nota: La hora local determina la zona horaria actual si no se establece la variables de entorno "TZ".

  c. Cree el directorio siguiente:

     ```
     mkdir -p /var/opt/OV/usr/share/lib
     ```

  d. Copie los archivos recursivamente de una carpeta a otra:

     ```
     cp -rp /usr/share/zoneinfo /var/opt/OV/usr/share/lib
     ```

  e. Cree el directorio llamado etc en /var/opt/OV ejecutando el siguiente comando:

     ```
     mkdir /var/opt/OV/etc
     ```

  f. Copie el archivo localtime en el directorio:

     ```
     cp /etc/localtime /var/opt/OV/etc
     ```

  g. Reinicie el agente ejecutando los siguientes comandos:

     i. `ovc -kill`

     ii. `ovc -start`

  Nota: ovbbccb usa la zona horaria actual cuando registra posteriormente una entrada en el archivo `System.txt`.

- *Problema:* **`ovcodautil -ds scope -o process -flat -rawonly`** genera un error de tiempo de espera expedido después de actualizar HPE Operations Agent de 11.14 a 12.01.

Causa: Las herramientas de generación de informes, como Performance Manager, no podrán crear gráficos de todos los datos de proceso seleccionados si el número de días es superior.

Solución: Reduce el número de métricas y su duración, y aumenta el tiempo de espera a un valor mayor.

Medida a tomar: Ejecute el siguiente comando para extraer los datos:

```
extract -xp -p
```

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 resolver esto, siga estos pasos:

  a. Inicie una sesión en el sistema como usuario raíz.
  b. Ejecute el comando siguiente para detener HPE 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:
     `chatr +pi 4K /opt/perf/bin/midaemon`
  e. Ejecute el comando siguiente para iniciar HPE Operations Agent:
     `/opt/OV/bin/opcagt -start`
- *Problema:* Tras instalar HPE 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.
- *Problema:* el siguiente mensaje de error aparece en la consola de HPOM:

  `CODA: GetDataMatrix returned 76='Method ScopeDataView::CreateViewEntity failed`

Causa: Este mensaje aparece si se usa el objeto `PROCESS` con la fuente de datos `SCOPE` en las directivas de umbral de medición en las que el origen se establece en el componente Embedded Performance.

Solución: Use la directiva de monitorización de servicio/proceso.

- *Problema: Los productos de análisis de datos, como HP Performance Manager o HPE Operations Bridge Reporter, no pueden recuperar datos del almacén de datos del agente y muestran el siguiente error:*

  ```
  Error al recuperar datos de la fuente de datos
  ```

  Causa: La utilidad de acceso a datos del agente lee todos los registros de una clase de datos con una consulta individual de un programa cliente. Las consultas las envían clientes de análisis de datos, como HP Performance Manager, al agente. Cuando una clase de datos contiene un alto volumen de registros, la utilidad de acceso a datos no puede procesar la consulta.

  Solución: Para evitar este problema, configure la utilidad de acceso a datos para que transfiera los registros de datos al cliente en varios segmentos. Siga estos pasos:

  a. Inicie sesión en el nodo del agente como usuario raíz o administrador.
  b. Ejecute el comando siguiente:

     En Windows:

     ```
     %ovinstalldir%bin\ovconfchg -ns coda -set DATAMATRIX_VERSION
     1
     ```

     En HP-UX/Linux/Solaris:

     ```
     /opt/OV/bin/ovconfchg -ns coda -set DATAMATRIX_VERSION 1
     ```

     En AIX:

     ```
     /usr/lpp/OV/bin/ovconfchg -ns coda -set DATAMATRIX_VERSION 1
     ```

     Ahora, en todas las consultas, la utilidad de acceso a datos del cliente descompone los datos en segmentos de cinco registros y, seguidamente, envía los datos al programa cliente. La descomposición de datos en segmentos mejora el rendimiento del proceso de transferencia de datos.

     Puede controlar el número de registros que el agente puede enviar al cliente con cada segmento. La variable `DATAMATRIX_ROWCOUNT` (disponible en el espacio de nombres `coda`) permite controlar este número (el valor predeterminado es cinco).

     La reducción del valor de la variable `DATAMATRIX_ROWCOUNT` puede aumentar marginalmente la velocidad de transferencia de datos cuando se introduzcan grandes volúmenes de datos en el almacén de datos.

Si en la variable `DATAMATRIX_ROWCOUNT` se selecciona 0, HPE Operations Agent vuelve a su comportamiento predeterminado de enviar registros de datos sin segmentos.

Sin embargo, se recomienda no cambiar la configuración predeterminada de la variable `DATAMATRIX_ROWCOUNT`.

c. Reinicie el agente para que los cambios surtan efecto.

```
ovc -restart coda
```

RTMA

- *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 resolver esto, siga estos pasos:

  a. Inicie una sesión en el sistema como usuario raíz.

  b. Ejecute el comando siguiente para detener HPE 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:

  ```
  chatr +pi 4K /opt/perf/bin/perfd
  ```

  e. Ejecute el comando siguiente para iniciar HPE Operations Agent:

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

GlancePlus

Problema: HP GlancePlus no muestra todas las instancias de LPAR hospedadas en un marco de AIX.

Causa: El LPAR donde ha instalado HPE Operations Agent no se puede comunicar con otros LPAR hospedados en el marco de AIX.

Solución: Asegúrese de que LPAR donde ha instalado HPE Operations Agent puede comunicarse con otros LPAR hospedados en el marco de AIX.

Ejecute el comando siguiente en el LPAR que hospeda el HPE Operations Agent para comprobar su conectividad con otros LPAR:

`xmpeek -l<nombre_host>`

En este ejemplo, *<nombre_host>* es el nombre de host de un LPAR.

hpsensor

- *Problema:* hpsensor que informa de un elevado uso de la memoria residente.

  Causa: Las estructuras de datos STL internas asignadas no se han limpiado después de la eliminación. Este problema aparece sobre todo en la plataforma HPUX.

  Solución: Para resolver este problema, siga estos pasos:

  a. Inicie sesión en el sistema como administrador.
  b. Vaya al directorio siguiente:
     - *En Windows*: `%OvDataDir%hpcs\`
     - *En UNIX/Linux*: `/var/opt/OV/hpcs/`
  c. Abra el archivo **hpcs.conf** y establezca la siguiente variable de configuración en **true** bajo el espacio de nombres **hpcs.runtime**:

     MemMap=<*true*>

Otro

- *Problema:* Las métricas `fs_type` informan un sistema autofs como NFS (sistema de archivos de red).

  Causa: En un sistema Linux donde HPE Operations Agent 12.01 se está ejecutando, un sistema autofs se informa como NFS si está en un kernel 2.6 o versiones inferiores.

  Solución: Para resolver este problema, asegúrese de que los sistemas autofs están en el kernel 3.0 o versiones superiores.

- *Problema:* El proceso **oacore** se detiene después de registrar el mensaje de error siguiente en el archivo `System.txt`:

  ```
  Database is in an inconsistent state.
  ```

  Causa: Si la información de la clase gestionada está presente sin el correspondiente archivo de base de datos, el proceso **oacore** se detiene después de registrar el mensaje de error en el archivo `System.txt`.

Nota: Ejemplo de un mensaje `System.txt` para la clase global:

```
0: INF: Thu Aug 20 16:14:28 2015: oacore (21580/139960775325472):
Collection intervals: Process = 60 Global = 300  DataFile Rollover%
= 20.

0: ERR: Thu Aug 20 16:14:28 2015: oacore (21580/139960775325472):
Database is in an inconsistent state. No database found for Class
Scope::Global

0: INF: Thu Aug 20 16:14:28 2015: oacore (21580/139960775325472):
(oacore-84) oacore. oacore Server stopped.
```

Solución: Para resolver este problema, debe suprimir el origen de datos o las clases específicas desde el origen de datos.

Utilice la herramienta `oadbutil.pl` para suprimir el origen de datos o las clases específicas desde el origen de datos.

Sintaxis:

- `oadbutil.pl -d <nombre_de_origen_de_datos>`

  Suprime el origen de datos junto con sus clases.

- `oadbutil.pl -d <nombre_de_origen_de_datos> -c <nombre_de_clase>`

  Suprime los metadatos y los registros solo de la clase especificada (se conserva la información sobre el origen de datos).

Por ejemplo:

```
oadbutil.pl –d Scope –c Global
```

En este ejemplo:

`<nombre_de_origen_de_datos>` es Scope

`<nombre_de_clase>` es Global

De manera predeterminada el recopilador de datos no crea un modelo para registrar la clase de métrica de rendimiento predeterminada y el origen de datos (Scope) en el Almacén de datos de métricas. Después de suprimir una clase de métrica de rendimiento o un origen de datos, siga los pasos para volver a crear el modelo:

a. Ejecute el comando siguiente para detener el proceso **oacore**:

   ```
   ovc -stop oacore
   ```

b. Ejecute el comando siguiente para recrear una clase:

```
ovconfchg -ns oacore -set UPDATED_MODEL_AVAILABLE true
```

c. Ejecute el comando siguiente para iniciar el proceso **oacore**:

```
ovc -start oacore
```

Nota: Si se suprime una clase personalizada (enviada a través de API que permiten ejecutar scripts basadas en Perl o DSI), se vuelve a crear automáticamente en un envío posterior.

Parte VII: Actualización a HPE Operations Agent versión 12.xx desde versiones anteriores

Con HPE Operations Agent versión 12.01, el Almacén de datos de métricas reemplaza el almacén de datos basado en el archivo de registro. Los procesos CODA y scope (scopeux en los nodos UNIX y Linux y scopent en los nodos Windows) se consolidan en un único proceso denominado **oacore**. El proceso **oacore** proporciona una interfaz de lectura y escritura para el rendimiento del sistema y los datos personalizados.

Los datos antiguos almacenados en los archivos de base de datos CODA, archivos de registros SCOPE y archivos de registro DSI se conservan en el modo de solo lectura. Puede acceder a los datos antiguos a través de utilidades como ovcodautil, extract o a través de herramientas de informes como HP Performance Manager y HP Reporter.

Nota:

- Los datos antiguos de HP Operations Agent 11.xx no migran al Almacén de datos de métricas.
- En un sistema HPUX IA, cuando se actualiza de HPE Operations Agent 11.xx a 12.xx, se hace una copia de seguridad de los archivos de base de datos antiguos y se guardan en:

  `/var/opt/OV/tmp/BackUp`

  No puede acceder a estos datos con herramientas como Extract, Utility o `ovcodautil`.
- En un sistema Linux o Windows, después de actualizar HPE Operations Agent 12.xx, no se puede acceder a los datos del sistema lógico (guardados en los archivos de registro) con herramientas como HP Performance Manager o HP Reporter.

Durante una actualización a HPE Operations Agent 12.xx desde versiones anteriores, todos los datos existentes en los archivos de registro DSI y los datos de Smart Plug-in (SPI) registrados en el almacén de datos CODA junto con todas las clases y métricas se crean automáticamente en el Almacén de datos de métricas.

Origen de datos de DSI

Antes de actualizar desde HPE Operations Agent 11.xx a 12.xx, asegúrese de que el origen de datos de DSI se agrega al `archivo datasources` ubicado en:

En HP-UX/Linux/Solaris:

`var/opt/OV/conf/perf`

En Windows:

`%ovdatadir%\conf\perf`

Ejecute el comando siguiente para agregar el origen de datos de DSI al archivo del almacén de datos:

`DATASOURCE=<Datasource Name>LOGFILE=<DSI Logfile Path>`

Asegúrese de que el nombre del archivo de registro y los nombres del origen de datos son iguales.

Por ejemplo:

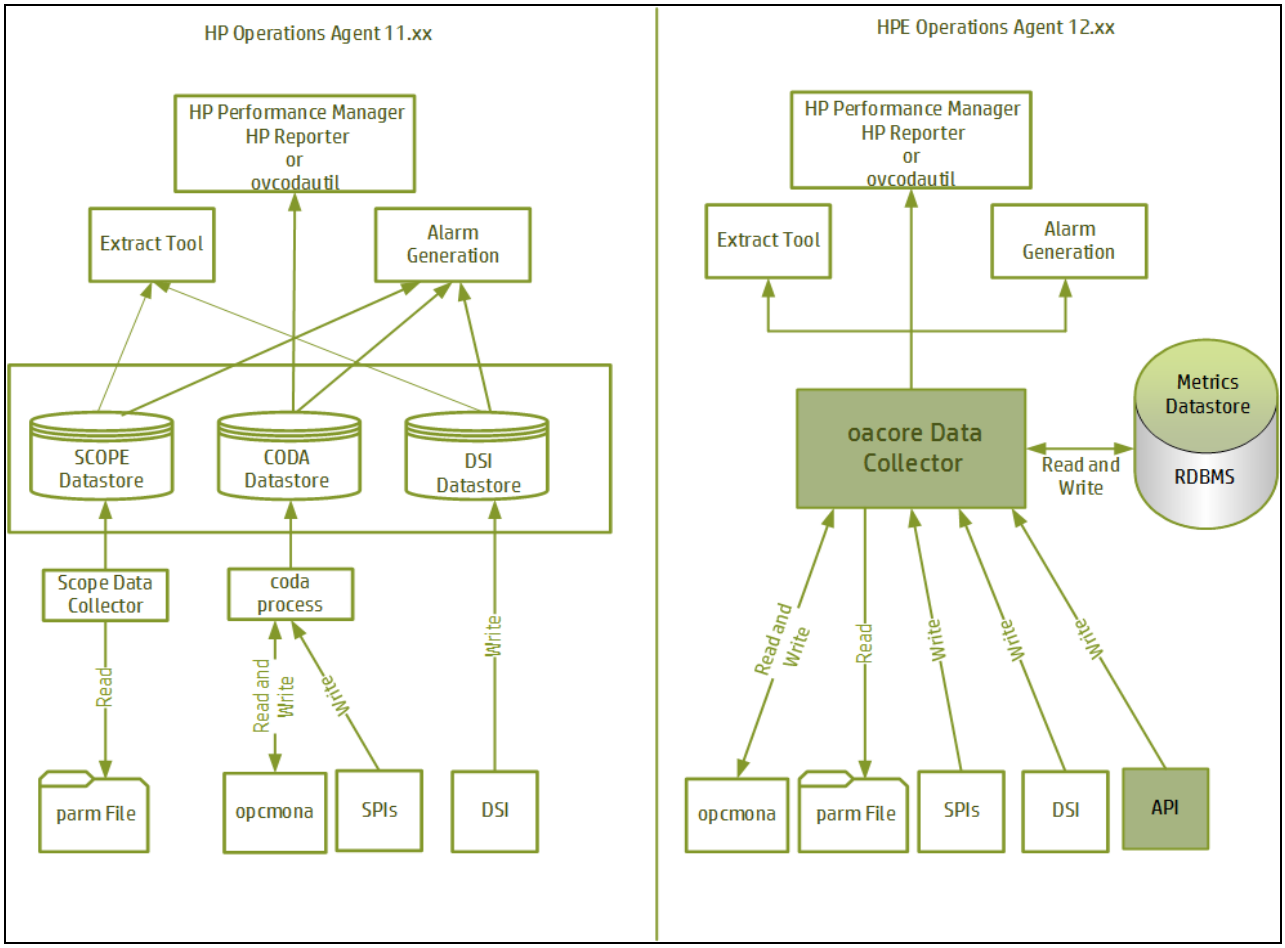
`DATASOURCE=`**`VMSTAT`** `LOGFILE=/root/dsi/`**`VMSTAT`**

El modelo para registrar los datos de DSI se crea en Almacén de datos de métricas solo si el origen de datos de DSI se agregan al `archivo de orígenes de datos`.

Nota: Cuando se registran los datos DSI con la opción `-timestamp` en HPE Operations Agent versión 11.xx, la marca de hora está en formato UTC. En HPE Operations Agent versión 12.xx, la marca de hora de los datos DSI registrados con la opción `-timestamp` se encuentra en formato de la zona horaria local.

Después de una actualización de HPE Operations Agent 11.xx a 12.xx, si exporta los datos DSI usando extraer, la marca de hora de los datos heredados (11.xx) estará en formato UTC y la de los nuevos datos (12.xx) estará en formato de la zona horaria local.

Comparación de HPE Operations Agent 12.xx con versiones anteriores



Comparación de HPE Operations Agent 12.xx con versiones anteriores

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
Recopilador de datos	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 el almacén de datos basado en el	El recopilador de datos **oacore** recopila continuamente los datos de estado y de rendimiento del sistema y almacena los datos recopilados en Almacén de datos de métricas.

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
	archivo de registro. Los datos recopilados se registran en los siguientes archivos de registro, dependiendo de las clases de datos **parm** file: logglob, logappl, logproc, logpcmd, logdev, logtran, logls y logindx.	**Nota:** Los procesos CODA y **scope** (Scopeux y Scopent) se consolidan en un único proceso denominado **oacore**. El proceso **oacore** proporciona una interfaz de lectura y escritura para el rendimiento del sistema y los datos personalizados.
Archivo `RUN` de **scope**	El archivo `RUN` de **scope** situado en `/var/opt/perf/datafiles/RUN` indica si **scope** se está ejecutando. Este archivo existe solo cuando **scope** (scopent y scopeux) se está ejecutando. Si **scope** finaliza de manera anómala, este archivo no se suprime. Para reiniciar **scope**, debe eliminar el archivo `RUN` de **scope** y después iniciar **scope**.	Este archivo no está presente. Para comprobar el estado del recopilador de datos **oacore**, consulte Verificación del estado del proceso oacore.
Almacén de datos	Los datos de métricas recopilados del sistema monitorizado se almacenan en el almacén de datos basado en el archivo de registro.	Los datos del estado y rendimiento del sistema recopilados de los sistemas monitorizados se almacenan en Almacén de datos de métricas. El Almacén de datos de métricas es un almacén de datos basado en el sistema de gestión de bases de datos relacionales

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
		(RDBMS). Los datos de métricas se recopilan en función de las clases de datos especificadas en el archivo **parm**. Para cada clase de datos que se registra en el almacén de datos de métricas, se crea un archivo de base de datos específico.

Orígenes de datos de proxy	Puede hospedar varios archivos de bases de datos de distintos sistemas como orígenes de datos. HP Operations Agent 11.xx puede leer simultáneamente de esos orígenes de datos.	Con HPE Operations Agent versión 12.xx, solo puede hospedar un origen de datos a la vez en modo de solo lectura.
		Siga estos pasos para hospedar los archivos de base de datos desde otro sistema como origen de datos:
		1. Ejecute el comando siguiente para detener el proceso **oacore**:
		`ovc -stop oacore`
		2. Realice una copia de seguridad de los archivos de base de datos existentes
		3. Ejecute el comando siguiente:
		`ovconfchg -ns oacore.dml -set READ_ONLY_MODE True`
		4. Coloque los archivos de

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx

		base de datos del sistema remoto en `datadir/databases/oa`.
		5. Ejecute el comando siguiente para iniciar el proceso **oacore**:
		`ovc -start oacore`
		Nota: Cuando el modo de proxy esté establecido, se deshabilita completamente el registro de datos en Almacén de datos de métricas.
		Siga los pasos para detener el hospedaje de archivos de base de datos de otro sistema como origen de datos:
		1. Ejecute el comando siguiente para detener el proceso **oacore**:
		`ovc -stop oacore`
		2. Ejecute el comando siguiente:
		`ovconfchg -ns oacore.dml -clear READ_ONLY_MODE`
		3. Ejecute el comando siguiente para iniciar el proceso **oacore**:
		`ovc -start oacore`

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
Proceso `rtmd`	El proceso `rtmd`, proporcionado por el componente RTM, permite establecer un canal de comunicación seguro para acceder a los datos en tiempo real desde el nodo.	El proceso `rtmd` se reemplaza por el proceso `hpsensor`. Las configuraciones de XPL para `rtmd` no son retrocompatibles y no funcionarán después de actualizar de HPE Operations Agent11.xx a 12.xx. El proceso `hpsensor` proporciona configuraciones de XPL similares para reforzar la seguridad (SSL). El proceso `hpsensor` ayuda a acceder a métricas de rendimiento en tiempo real a través de un canal de comunicación seguro, de forma local o remota.

Componente Performance Collection

Comparación de HPE Operations Agent 12.xx con versiones anteriores

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
Interfaz gráfica de usuario del componente Performance Collection	Puede usar la interfaz gráfica de usuario del componente Performance Collection para realizar las tareas siguientes: extraer datos de archivo de registro, exportar datos de archivo de registro, archivar datos de archivo de registro, cambiar el tamaño del archivo de registro, examinar un archivo de registro,	No se admite la interfaz gráfica de usuario del componente Performance Collection.

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
	configurar plantillas de exportación, configurar opciones de usuario, configurar parámetros de recopilación, configurar definiciones de alarma y comprobar el estado.	

`ovcodautil` es una utilidad de línea de comandos usada para mostrar las métricas registradas.	`ovcodautil -ds SCOPE -o Global -m GBL_CPU_TOTAL_UTIL`
	`/opt/OV/bin/ovcodautil [options]`
	`-h`: Muestra el encabezado de métrica
		`ovcodautil -ds SCOPE -o Global -m GBL_CPU_TOTAL_UTIL`
		`/opt/OV/bin/ovcodautil [options]`
		`-header`: Muestra el encabezado de métrica
		`-h`: Muestra el mensaje de ayuda

Archivo parm

Comparación de HPE Operations Agent 12.xx con versiones anteriores

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
Transacciones de ámbito	El mismo recopilador **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_**	Con HPE Operations Agent 12.xx, Scope_Get_Process_Metrics y Scope_Get_Global_Metrics no se registran.

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
	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**.	
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. **Nota:** 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.	El parámetro `Mainttime` no se admite con HPE Operations Agent versión 12.xx. El parámetro *size* se utiliza para establecer el tamaño máximo (en megabytes) de los archivos de base de datos. Los archivos de base de datos que se almacenan en las clases de métrica de rendimiento predeterminadas se sustituyen cuando se alcanza el tamaño máximo especificado en el archivo **parm**. Si se cambia la especificación del tamaño del archivo **parm**, **oacore** lo detecta durante el inicio.

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
		Si el tamaño no se especifica en el archivo **parm**, los archivos de base de datos se sustituyen cuando se alcanza el tamaño máximo de 1GB. Durante una sustitución, el 20 por ciento de los datos más antiguos se suprimirán de los archivos de base de datos.
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	El parámetro *days* no se admite con HPE Operations Agent versión 12.xx. El parámetro *size* se utiliza para establecer el tamaño máximo (en megabytes) de los archivos de base de datos. Los archivos de base de datos que se almacenan en las clases de métrica de rendimiento predeterminadas se sustituyen cuando se alcanza el tamaño máximo especificado en el archivo **parm**. Si se cambia la

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
	`maintime`. Durante la reversión, los datos recopilados después del parámetro days que hayan alcanzado su valor máximo se suprimirán del archivo de registro. **Nota:** Si los archivos de registro se revierten 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.	especificación del tamaño del archivo **parm**, **oacore** lo detecta durante el inicio. Si el tamaño no se especifica en el archivo **parm**, los archivos de base de datos se sustituyen cuando se alcanza el tamaño máximo de 1GB. Durante una sustitución, el 20 por ciento de los datos más antiguos se suprimirán de los archivos de base de datos.
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	El parámetro *Maintweekday* no se admite con HPE Operations Agent versión 12.xx. El parámetro *size* se utiliza para establecer el tamaño máximo (en megabytes) de los archivos de base de

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
	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. **Nota:** 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	datos. Los archivos de base de datos que se almacenan en las clases de métrica de rendimiento predeterminadas se sustituyen cuando se alcanza el tamaño máximo especificado en el archivo **parm**. Si se cambia la especificación size del archivo **parm**, **oacore** lo detecta durante el inicio. Si el tamaño no se especifica en el archivo **parm**, los archivos de base de datos se sustituyen cuando se alcanza el tamaño máximo de 1GB. Durante una sustitución, el 20 por ciento de los datos más antiguos se suprimirán de los archivos de base de datos.

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
	`maintime`.	
javaarg	Este parámetro sólo es válido en UNIX/Linux.	Este parámetro es válido en las plataformas Windows y UNIX.
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.	Este parámetro se admite en todas las plataformas. El parámetro `proccmd` permite el registro de los comandos de proceso en el almacén de datos de . De manera predeterminada, el valor de este proceso se establece en 0 y se deshabilita el registro de los comandos del proceso. Para habilitar el registro de los comandos del proceso, establezca el valor de este parámetro en 1. El registro del parámetro `proccmd` se activa cuando el valor de este parámetro es mayor o igual que 1. La longitud del comando del proceso registrado es siempre 4096, con independencia del valor especificado en este parámetro.

Programa Utility

Comparación de HPE Operations Agent 12.xx con versiones anteriores

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
Ejecución del programa `utility`	HP Operations Agent 11.xx admite el modo por lotes, el modo interactivo y el modo de línea de comandos para ejecutar el programa `utility`.	HPE Operations Agent 12.xx solo admite el modo de línea de comandos para ejecutar el programa `utility`.
Comando resize	Use el comando resize `-xr` para gestionar el espacio de su conjunto de archivos de registro. 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.	El comando resize no se admite con HPE Operations Agent versión 12.xx. El parámetro *size* se utiliza para establecer el tamaño máximo (en megabytes) de los archivos de base de datos. Los archivos de base de datos que se almacenan en las clases de métrica de rendimiento predeterminadas se sustituyen cuando se alcanza el tamaño máximo especificado en el archivo **parm**. Si se cambia la especificación del tamaño del archivo **parm**, **oacore** lo detecta durante el inicio. Durante una sustitución, el 20 por ciento de los datos más antiguos se suprimirán de los archivos de base de datos que excedan del tamaño especificado en el archivo **parm**.
Comandos del programa	Se admiten estos comandos y puede usarlos de la forma	Estos comandos no se admiten.

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx

utility:
`start`
`stop`
`exit`
`guide`
`logfile`
`menu`
`sh`
`show`
	siguiente:
	• `start` - Especifica la fecha de inicio y la hora de una función de exploración o de análisis.
	• `stop` - Especifica la fecha y la hora término de una función de exploración o de análisis.
	• `exit` - El comando `exit` permite finalizar el programa **utility**.
	• `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.
	• `logfile` - El comando `logfile` permite abrir un archivo de registro.
	• `menu`: use el comando `menu` para imprimir una lista de los comandos de utility disponibles.
	• `sh`: use `sh` para introducir un comando shell sin salir de utility.
	• `show`: el comando `show` enuncia los nombres de los archivos que están abiertos y el estado de los	

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

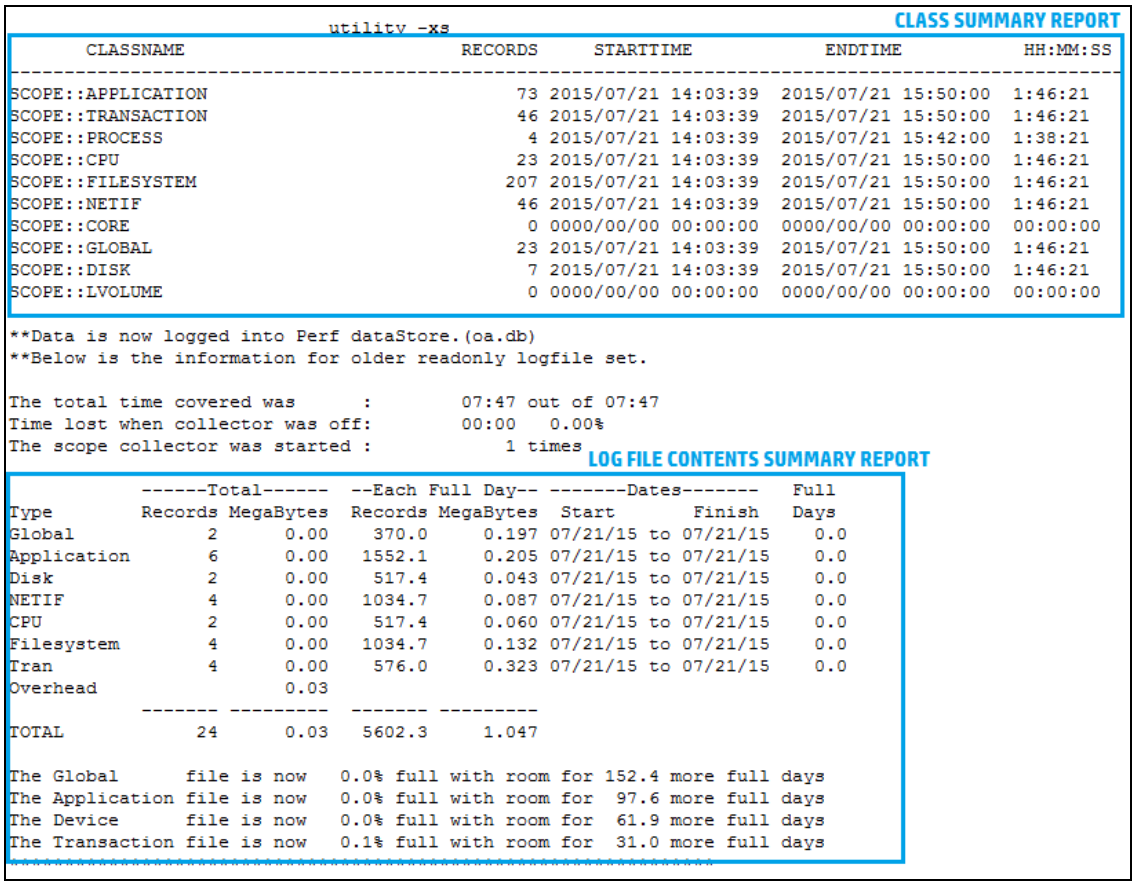
Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
	parámetros utility que pueden establecerse.	

Informe de exploración del programa utility

Si actualiza desde HP Operations Agent 11.xx a HP Operations Agent 12.xx, el informe de exploración del programa utility proporciona la información sobre el espacio de disco utilizado por cada clase de datos, así como información sobre los datos heredados almacenados en el conjunto de archivo de registro.

Por ejemplo:

Si ejecuta el comando `utility -xs`, se generan el informe de resumen de clase y el informe de resumen de contenidos del archivo de registro, como se muestra en la figura siguiente:



```
                        utility -xs                                  CLASS SUMMARY REPORT
      CLASSNAME                          RECORDS     STARTTIME           ENDTIME           HH:MM:SS
-------------------------------------------------------------------------------------------------------
SCOPE::APPLICATION                            73 2015/07/21 14:03:39  2015/07/21 15:50:00  1:46:21
SCOPE::TRANSACTION                            46 2015/07/21 14:03:39  2015/07/21 15:50:00  1:46:21
SCOPE::PROCESS                                 4 2015/07/21 14:03:39  2015/07/21 15:42:00  1:38:21
SCOPE::CPU                                    23 2015/07/21 14:03:39  2015/07/21 15:50:00  1:46:21
SCOPE::FILESYSTEM                            207 2015/07/21 14:03:39  2015/07/21 15:50:00  1:46:21
SCOPE::NETIF                                  46 2015/07/21 14:03:39  2015/07/21 15:50:00  1:46:21
SCOPE::CORE                                    0 0000/00/00 00:00:00  0000/00/00 00:00:00  00:00:00
SCOPE::GLOBAL                                 23 2015/07/21 14:03:39  2015/07/21 15:50:00  1:46:21
SCOPE::DISK                                    7 2015/07/21 14:03:39  2015/07/21 15:50:00  1:46:21
SCOPE::LVOLUME                                 0 0000/00/00 00:00:00  0000/00/00 00:00:00  00:00:00

**Data is now logged into Perf dataStore.(oa.db)
**Below is the information for older readonly logfile set.

The total time covered was      :        07:47 out of 07:47
Time lost when collector was off:        00:00   0.00%
The scope collector was started :            1 times
                                                  LOG FILE CONTENTS SUMMARY REPORT
            ------Total------  --Each Full Day-- -------Dates-------   Full
Type        Records MegaBytes  Records MegaBytes  Start       Finish   Days
Global            2      0.00    370.0     0.197 07/21/15 to 07/21/15    0.0
Application       6      0.00   1552.1     0.205 07/21/15 to 07/21/15    0.0
Disk              2      0.00    517.4     0.043 07/21/15 to 07/21/15    0.0
NETIF             4      0.00   1034.7     0.087 07/21/15 to 07/21/15    0.0
CPU               2      0.00    517.4     0.060 07/21/15 to 07/21/15    0.0
Filesystem        4      0.00   1034.7     0.132 07/21/15 to 07/21/15    0.0
Tran              4      0.00    576.0     0.323 07/21/15 to 07/21/15    0.0
Overhead                 0.03
            ------- ---------  ------- ---------
TOTAL            24      0.03   5602.3     1.047

The Global      file is now   0.0% full with room for 152.4 more full days
The Application file is now   0.0% full with room for  97.6 more full days
The Device      file is now   0.0% full with room for  61.9 more full days
The Transaction file is now   0.1% full with room for  31.0 more full days
```

El informe de resumen de contenidos del archivo de registro muestra la información almacenada en el conjunto de archivos de registro de solo lectura más antiguos.

Por ejemplo: Si ejecuta el comando `utility -xs -D`, se generan los informes siguientes:

Información global inicial del archivo **parm**

HP Operations Agent 11.xx

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 una sección de un informe en el que se indica el contenido del archivo `parm`.

```
04/01/2014 12:38 System ID="  "
SCOPE/UX C.05.00.100 COLLECTION INTERVALS: Process = 60, GLOBAL = 300
Seconds, Log version=D
Configuration: ia64, O/S Linux 2.6.16.21-#1 SMP Monia64 #1 SMP Monia64
CPUs=1
Logging Global(NoDisk)(NoNETIF) Application Process Transaction
        Device = Disk NETIF FileSys CPU records

Thresholds: CPU=10.00%, Disk=n/a, ProcMem=900.0 MB
Nonew=TRUE, Nokilled=TRUE, Shortlived=FALSE (<1 sec)

Javaarg = true

Parms: Buffer Cache Size = 0KB, NPROC = 0

Memory:  Physical = 981 MB, Swap = 1286.4 MB,
Available to users = 853 MB

Application and device data flush frequency = 3600 seconds
01/02/2014 09:22 Data collected on 3 disk devices:
    Disk #0              = "sda"
    Disk #2              = "sdb"
    Disk #3              = "hda"
```

En esta instancia, 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 el

recopilador de datos. 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*.

HPE Operations Agent 12.xx

Este informe se genera solo si se actualiza de HPE Operations Agent 11.xx a 12.01.

Definiciones de aplicación iniciales del archivo **parm**

HP Operations Agent 11.xx

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:

```
10/29/2013 15:25 Application(1) = "other"
Comment=all processes not in user-defined applications

10/29/2013 15:25 Application(2) = "network"
File=nfs*,biod,automount,amd,*inetd,snmp*,rpc*,llbd,netfmt,portmap
File=rbootd,telnet*,ftp*,*rlogin*,remsh*

File=rcp,nktl*,nvsisr,ttisr,lcsp,gcsp,strmen,strweld,vtdaemon
File=mib*,trapdest*,*web*,xntpd,yp*

File=hp_unixagt,ntl*,pty*

10/29/2013 15:25 Application(3) = "xwindows"
File=X*,xload,xclock,*term,grmd,softmsg*,vue*,ttsession,pexd
File=dt*,xfs,rpc.ttdbserver,Aserver,gdm*,kdm*

File=gnome*,kde*,soffice.bin,netscape*,mozilla*,realplay

10/29/2013 15:25 Application(4) = "memory_management"
File=swapper,vhand,syncer,pageout,fsflush,kswapd,kreclaimd,bdflush
```

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.

HPE Operations Agent 12.xx

Este informe se genera solo si se actualiza de HPE Operations Agent 11.xx a 12.01.

notificaciones de cambios globales del archivo **parm**

HP Operations Agent 11.xx

Este informe se genera 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*"
```

HPE Operations Agent 12.xx

Este informe no se genera.

Cambios de aplicación del archivo **parm**

HP Operations Agent 11.xx

Para obtener el informe de cambios de aplicación del archivo de `parm`, use el comando `scan` con su `detail on` predeterminado y asegúrese de tener datos de aplicación en el almacén de datos.

Cualquier cambio realizado en el archivo `parm` requiere que **scope** se reinicie para registrar el cambio en el almacén de datos. 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
```

Nota: 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 a acompañada de un cambio de nombre.

HPE Operations Agent 12.xx

Este informe no se genera.

Notificaciones de tiempo de inactividad de `scope`.

HP Operations Agent 11.xx

Este informe indica cuándo `scope` se ha reiniciado y cuándo los últimos datos válidos se registraron en el archivo de registro antes de que se reiniciara scope.

HPE Operations Agent 12.xx

Este informe no se genera.

Informes de resúmenes específicos de aplicación

HP Operations Agent 11.xx

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.

El informe de resumen específico de la aplicación se genera cuando cambien las definiciones de la aplicación. Esto permite al usuario ver las definiciones de la aplicación antes y después de que se produzca el cambio. Se generará un informe final por aplicación. Este informe solo comprende el tiempo transcurrido desde que se generó el último informe y no el tiempo total que cubre el archivo de registro.

Por ejemplo:

```
 Percent of Total

Application          Records      CPU   Disk  Trans

-------------------- ---------- ------ ------ ------

other                    26258   3.7%   6.6% 100.0%
network                   4649   0.0%   0.4%   0.0%
xwindows                 34571  25.3%   1.9%   0.0%
```

```
memory_management          3797   0.0%   0.0%   0.0%
other_user_root           45504  71.0%  91.0%   0.0%

------------------- ---------- ------ ------ ------

All user applications     77.1%  96.3%  93.4%   0.0%
```

HPE Operations Agent 12.xx

Este informe se genera solo si se actualiza de HPE Operations Agent 11.xx a 12.01.

Informe de resumen de proceso

HP Operations Agent 11.xx

Este informe se imprime siempre que los datos de proceso sean explorados. Para obtener este informe se deberán tener datos de proceso en el archivo de datos.

Este informe le ayuda a establecer los umbrales de proceso *interesantes* para `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:  11/08/2013 10:18    to 05/05/2014 16:55
There were 3427.4 hours of process data
Process records were logged for the following reasons:

Log Reason      Records   Percent  Recs/hr
--------------- -------   -------  -------
New Processes         6      0.1%      0.0
Killed Processes      6      0.1%      0.0
CPU Threshold      6956    100.0%      2.0

NOTE: A process may 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, solose generará un informe que cubre el periodo explorado en su totalidad.

Para reducir la cantidad de datos de proceso registrados por `scope`, modifique el parámetro `threshold` del archivo parm y aumente 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.

HPE Operations Agent 12.xx

Este informe no se genera.

Inicio y detención de la exploración

HP Operations Agent 11.xx

El informe de inicio y detención de la exploración se imprimirá si se han explorado datos válidos. Ofrece las fechas y horas reales en las que se inició y detuvo la exploración.

Por ejemplo:

```
Scan started on 08/11/2013 10:18 PM
Scan stopped on 05/05/2014 16:55
```

HPE Operations Agent 12.xx

Este informe se genera solo si se actualiza de HPE Operations Agent 11.xx a 12.01.

Resumen general de aplicación

HP Operations Agent 11.xx

Para obtener este informe se deberán tener datos de aplicación en el almacén de datos HPE Operations Agent.

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 significante 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
88521 out of 114779 records ( 77.1%)
832.2 out of 863.9 CPU hours ( 96.3%)
819.2 out of 877.2 M disk IOs ( 93.4%)
```

```
 0.0 out of 0.0 M trans ( 0.0%)
```

HPE Operations Agent 12.xx

El informe resumen general de aplicación solo se genera si se actualiza de HPE Operations Agent 11.xx a 12.01. Este informe solo contiene datos heredados de los archivos de base de datos de CODA, archivos de registro de `scope` y archivos de registro de DSI.

Resumen de cobertura del recopilador

HP Operations Agent 11.xx

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`.

Por ejemplo:

```
The total time covered was : 172/13:29:27 out of 178/06:37:00
Time lost when collector was off: 5/17:07:33 3.20%
The scope collector was started : 5 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).

HPE Operations Agent 12.xx

Este informe se genera solo si se actualiza de HPE Operations Agent 11.xx a 12.01.

Nota: Los detalles de inicio y detención del proceso **oacore** se enumeran en el archivo `System.txt`.

Resumen de clase y resumen de contenidos del archivo de registro

HP Operations Agent 11.xx

El informe de resumen de clase no se admite con HPE Operations Agent 11.xx.

El informe de 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`.

Por ejemplo:

```
            ------Total------  --Each Full Day-- -------Dates-------   Full
Type        Records MegaBytes  Records MegaBytes  Start       Finish   Days
Global         7460      3.97   1440.1     0.766 07/03/2015 to 07/08/2015    5.2
Application    9161      1.21   1768.7     0.233 07/03/2015 to 07/08/2015    5.2
Process       14920      4.45   2880.5     0.858 07/03/2015 to 07/08/2015    5.2
Disk           7451      0.63   1438.5     0.121 07/03/2015 to 07/08/2015    5.2
NETIF         14920      1.25   2880.5     0.242 07/03/2015 to 07/08/2015    5.2
CPU           29840      3.46   5761.1     0.668 07/03/2015 to 07/08/2015    5.2
Filesystem    37300      4.77   7201.3     0.922 07/03/2015 to 07/08/2015    5.2
Tran          14921      8.36   2880.2     1.613 07/03/2015 to 07/08/2015    5.2
Overhead                 0.58
            ------- ---------  ------- ---------
TOTAL        135973     28.67  26250.9     5.424

The Global      file is now  13.3% full with room for  33.9 more full days
The Application file is now   6.3% full with room for  80.3 more full days
The Process     file is now  15.0% full with room for  29.7 more full days
The Device      file is now  52.4% full with room for   4.9 more full days
The Transaction file is now  84.2% full with room for   1.0 more full days
```

HPE Operations Agent 12.xx

El informe de resumen de clase solo se genera en una instalación nueva de HPE Operations Agent 12.01. Proporciona la información sobre el espacio de disco utilizado por cada clase de datos.

Si actualiza de HPE Operations Agent 11.xx a 12.01 se general los informes de resumen de clase y de resumen de contenidos del archivo de registro.

Resumen de espacio libre del archivo de registro

HP Operations Agent 11.xx

Este informe se imprimirá con cada archivo de registro explorado.

Por ejemplo:

```
The Global      file is now  73.9% full with room for   0.3 more full days
The Application file is now  78.2% full with room for  21.8 more full days
The Process     file is now   7.0% full with room for  93.0 more full days
The Device      file is now  96.4% full with room for   3.1 more full days
The Transaction file is now  90.4% full with room for   3.0 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.

HPE Operations Agent 12.xx

El informe de resumen del espacio vacío del archivo de registro solo se genera si actualiza de HPE Operations Agent 11.xx a 12.01. Este informe solo contiene datos heredados de los archivos de base de datos de CODA, archivos de registro de `scope` y archivos de registro de DSI.

Extract

Comparación de HPE Operations Agent 12.xx con versiones anteriores

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
Ejecución del programa `extract`	HP Operations Agent 11.xx admite el modo interactivo y el modo de línea de comandos para ejecutar el programa `extract`.	HPE Operations Agent 12.xx solo admite el modo de línea de comandos para ejecutar el programa `extract`.
Archivos de salida	El programa `extract` realiza la función de exportación. Lee los datos de los archivos de registro y exporta los resultados a los archivos de salida en formato ASCII, archivo de datos, binario y WK1 (hoja de cálculo).	El programa Extract realiza la función de exportación. Lee los datos del almacén de datos y exporta los resultados a los archivos de salida en formato ASCII.
Adaptador de bus de host (HBA)	Esta clase de métrica no está disponible en HP Operations Agent 11.xx.	Se agrega una nueva clase de métrica, el adaptador de bus de host (HBA). Use los comandos siguientes para exportar los datos de HBA: • `-h` - Especifica los datos de detalle de HBA que se van a

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
		exportar. • `-H` - Especifica los datos de resumen de HBA que se van a exportar.

Comandos de extract:
`-s`
`-k`
`-ut`
`-v`
`-we`
`-xt`
`-xw`
`-xm`
`-xy`
	Se admiten estos comandos y puede usarlos de la forma siguiente:
	• `-s` - Especifica la fecha de inicio y de fin para períodos específicos excluidos los fines de semana.
	• `-k` - Exporta sólo procesos cerrados.
	• `-ut` - Muestra la fecha y la hora en formato UNIX en los datos de archivo de registro exportados.
	• `-v` - Genera formatos de informes de salida detallada.
	• `-we` - Especifica días que se deben excluir de la exportación; 1=domingo.
	• `-xt` - Extrae datos en un formato interno del sistema
	• `-xw` - Extrae datos de una semana natural
	• `-xm` - Extrae datos de un mes natural
	• `-xy` - Extrae datos de un año natural
		Estos comandos no se admiten.

Opciones de archivo:	Se admiten	No admitido

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
Append, Purge,New		
Opciones de exportación: TODAY, TOMORROW, TODAY-1 TOMORROW-1, Last, First	Se admiten	Solo se admiten las opciones última y primera cuando se usan con las opciones de comando `-b` y `-e`.
Formato de datos exportados: ASCII, DATAFILE, WK1 o SPREADSHEET, BINARY	Se admiten	Solo se admite el formato ASCII
Archivos de plantilla de exportación: Repthist, Reptall	Se admiten	Solo se proporciona el archivo de plantilla Reptall.
Diseño: único, múltiple	Se admiten	Solo se admite un único diseño.
Título del archivo de exportación: !date, !time, !logfile, !class, !collector, !system_id	Se admiten	No admitido

Métrica

Comparación de HPE Operations Agent 12.xx con versiones anteriores

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
BLANK RECORD_ TYPE DATE TIME YEAR DAY DATE_ SECONDS INTERVAL STATDATE STATTIME	Estas métricas se admiten.	Estas métricas no se admiten por ninguna clase de datos.
Métricas de seguimiento de transacciones	Las métricas de seguimiento de transacciones incluyen las métricas correspondiente a todas las transacciones del sistema realizadas en el sistema monitorizado. Las métricas pertenecientes a esta clase van precedidas del prefijo TTBIN_ o TT_.	Las métricas de seguimiento de transacciones incluyen las métricas correspondiente a todas las transacciones del sistema realizadas en el sistema monitorizado. Las métricas pertenecientes a esta clase van precedidas del prefijo TT_. Las métricas siguientes no se admiten: TT_USER_MEASUREMENT_ NAME TT_USER_MEASUREMENT_ MAX

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

		TT_USER_MEASUREMENT_MIN TT_USER_MEASUREMENT_AVG TT_USER_MEASUREMENT_NAME_2 TT_USER_MEASUREMENT_MAX_2 TT_USER_MEASUREMENT_MIN_2 TT_USER_MEASUREMENT_AVG_2 TT_USER_MEASUREMENT_NAME_3 TT_USER_MEASUREMENT_MAX_3 TT_USER_MEASUREMENT_MIN_3 TT_USER_MEASUREMENT_AVG_3 TT_USER_MEASUREMENT_NAME_4 TT_USER_MEASUREMENT_MAX_4 TT_USER_MEASUREMENT_MIN_4 TT_USER_MEASUREMENT_AVG_4 TTBIN_UPPER_RANGE_10 TTBIN_UPPER_RANGE_1 TTBIN_UPPER_RANGE_2 TTBIN_UPPER_RANGE_3

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

		TTBIN_UPPER_RANGE_4 TTBIN_UPPER_RANGE_5 TTBIN_UPPER_RANGE_6 TTBIN_UPPER_RANGE_7 TTBIN_UPPER_RANGE_8 TTBIN_UPPER_RANGE_9 TTBIN_TRANS_COUNT_10 TTBIN_TRANS_COUNT_1 TTBIN_TRANS_COUNT_2 TTBIN_TRANS_COUNT_3 TTBIN_TRANS_COUNT_4 TTBIN_TRANS_COUNT_5 TTBIN_TRANS_COUNT_6 TTBIN_TRANS_COUNT_7 TTBIN_TRANS_COUNT_8 TTBIN_TRANS_COUNT_9 TT_NUM_BINS
Adaptador de bus de host (HBA)	Esta clase de métrica no está disponible en HP Operations Agent 11.xx.	HBA es la nueva clase de métricas que se registra junto con otras clases de métricas. Incluye la métrica correspondiente a todos los adaptadores del bus de host que están siendo ejecutados en el sistema de monitorización. Las métricas pertenecientes a esta clase van precedidas del prefijo BYHBA_. Las métricas siguientes se registran:

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

		BYHBA_TIME BYHBA_INTERVAL BYHBA_ID BYHBA_NAME BYHBA_DEVNAME BYHBA_DEVNO BYHBA_CLASS BYHBA_DRIVER BYHBA_STATE BYHBA_UTIL BYHBA_THROUGHPUT_UTIL BYHBA_IO BYHBA_READ BYHBA_WRITE BYHBA_IO_RATE BYHBA_READ_RATE BYHBA_WRITE_RATE BYHBA_BYTE_RATE BYHBA_READ_BYTE_RATE BYHBA_WRITE_BYTE_RATE BYHBA_REQUEST_QUEUE BYHBA_BUSY_TIME BYHBA_AVG_WAIT_TIME BYHBA_AVG_SERVICE_TIME BYHBA_TYPE
Métricas BYLS	Estas métricas se admiten.	Las métricas BYLS no se admiten en las plataformas

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

		Windows y Linux. HPE Operations Agent 12.xx no recopila las métricas BYLS de Xen, KVM, VMware vSphere, Hyper-V y otros dominios de virtualización.

Interceptor de capturas de SNMP

Comparación de HPE Operations Agent 12.xx con versiones anteriores

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
El proceso **opctrapi**	El proceso **opctrapi** está configurado para interceptar las capturas de SNMPv1 y SNMPv2.	El proceso **opctrapi** está configurado para interceptar las capturas de SNMPv1, SNMPv2, SNMPv3 y los mensajes de información.
Variable SNMP_ SESSION_ MODE	La variable SNMP_SESSION_MODE le permite configurar el interceptor de capturas de SNMP para que use distintos mecanismos para conectarse al puerto 162 para que escuche capturas de SNMP cuyo origen se encuentre en fuentes externas. • Para configurar el interceptor de capturas de SNMP para conectarse al puerto 162 usando la	La variable SNMP_ SESSION_ MODE no se admite. Con HPE Operations

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx

	API Net-SNMP para escuchar capturas de SNMP, ejecute el siguiente comando:
	`ovconfchg -ns eaagt set SNMP_SESSION_MODE NETSNMP`
	• Sólo en nodos de Windows. Para configurar el interceptor de capturas de SNMP para usar el servicio de captura de Microsoft para escuchar capturas de SNMP, ejecute el siguiente comando:
	`ovconfchg -ns eaagt set SNMP_SESSION_MODE WIN_SNMP`
	• Sólo en nodos de UNIX/Linux. Para configurar el interceptor de capturas de SNMP para conectarse al puerto 162 usando la API OVSNMP para escuchar capturas de SNMP, ejecute el siguiente comando:
	`ovconfchg -ns eaagt set SNMP_SESSION_MODE NO_TRAPD`
	• Para configurar el interceptor de capturas de SNMP para conectarse al puerto 162 usando la API OVSNMP para escuchar capturas de SNMP, ejecute el siguiente comando:
	`ovconfchg -ns eaagt set SNMP_SESSION_MODE NNM_LIBS`
	• Para configurar el interceptor de capturas de SNMP para intentar suscribirse al demonio PMD de NNM (7.5x) para escuchar capturas de SNMP, ejecute el siguiente comando:
	`ovconfchg -ns eaagt set SNMP_SESSION_MODE TRY_BOTH`
	• Sólo en nodos de UNIX/Linux. Para configurar el
		Agent 12.xx solo se admite el modo NETSNMP.

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
	interceptor de capturas de SNMP para suscribirse al demonio PMD de NNM (7.5x) para escuchar capturas de SNMP, ejecute el siguiente comando: `ovconfchg -ns eaagt set SNMP_SESSION_ MODE NNM_PMD`	

Integración de orígenes de datos

Comparación de HPE Operations Agent 12.xx con versiones anteriores

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
Recopilación de datos	Los datos recopilados por DSI se almacenan en archivos de registro.	Los datos recopilados por DSI se almacenan en Almacén de datos de métricas. Para cada clase de datos que se registra en el almacén de datos, se crea un archivo de base de datos específico. **Nota:** Para que DSI recopile datos personalizados, **oacore** debe estar ejecutándose. Sin embargo, para conservar la compatibilidad con versiones anteriores, la línea de comandos sigue admitiendo el argumento del archivo de registro. El nombre del archivo de registro se extrae de la ruta y se considera como el nombre del origen de datos. `sdlcomp <class specication file> <logfile name>`

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
		Cuando actualice de HPE Operations Agent 11.xx a 12.xx, los metadatos o el formato para almacenar datos se copia de los archivos de registro antiguos al Almacén de datos de métricas. **Nota:** El compilador de DSI **sdlcomp** no creará archivos basados en el nombre del archivo de registro; por el contrario, el nombre del archivo de registro se usa como nombre de origen de datos. Para acceder a los datos antiguos registrados en los archivos de registro de DSI, consulte "Orígenes de datos de DSI".
Sustitución de datos	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	`INDEX BY, MAX INDEXES` y `ROLL BY` no se admiten. El tamaño máximo de los archivos de base de datos que almacenan los *datos personalizados* se establece en 1 GB de manera predeterminada. Este tamaño no se puede configurar. Los datos almacenados en el almacén de datos de métricas se sustituyen automáticamente cuando los archivos de base de datos alcanzan un tamaño máximo de 1 GB. Durante una sustitución, el veinte por ciento de los datos más antiguos se suprimirán (el archivo más antiguo de la base de datos se elimina de manera permanente).

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
	bloques a almacenar y el tamaño de los bloques de datos que serán suprimidos cuando los datos alcancen su valor de índice máximo.	
ACTION	El comando mencionado con el parámetro ACTION se ejecutará antes de la sustitución.	Este parámetro no se admite.
CAPACITY	CAPACITY es el número de registros que serán almacenados en la clase.	No admitido
Sintaxis de sdlutil	`sdlutil logfile_set [opción]` En este ejemplo, logfile_set es el nombre de un conjunto de	`sdlutil <nombre de archivo de registro> [opción]` Variables y opciones: • -rm all Suprime los orígenes de datos, clases, métricas y datos del almacén de datos de métricas para el origen de datos especificado.

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
	archivos de registro creado mediante la compilación de una especificación de clase. Variables y opciones: • -classes classlist Proporciona una descripción de clase de todas las clases enumeradas. Si no se enumera ninguna, se proporcionarán todas. Separe los ítems en la classlist con espacios. • -stats classlist Proporciona estadísticas completas de todas las clases	• -vers Muestra información de la versión. • -? Muestra la descripción de sintaxis.

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
	enumeradas. Si no se enumera ninguna, se proporcionarán todas. Separe los ítems en la classlist con espacios. • -metrics metriclist Ofrece descripciones de métrica de toda la métrica contenida en la metriclist. Si no se enumera ninguna, se proporcionarán todas. Separe los ítems en la metriclist con espacios. • -id Muestra el ID del segmento de memoria compartida	

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
	que utiliza el archivo de registro. -files enumera todos los archivos del conjunto de archivos de registro. • -rm all Suprime todas las clases y datos así como sus datos y el ID de memoria compartida del archivo de registro. • -decomp classlist Vuelve a crear una especificación de clase a partir de la información del conjunto de archivos de registro. Los resultados	

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
	se escriben en stdout y se deben redireccionar a un archivo si se prevé realizar cambios en el archivo y utilizarlo de nuevo. Separe los ítems en la classlist con espacios. • -vers Muestra información de la versión. • -? Muestra la descripción de sintaxis.	
PRECISION	`PRECISION` identifica el número de posiciones decimales que será usado con los valores de métrica.	No admitido

Comparación de HPE Operations Agent 12.xx con versiones anteriores, continuación

Novedades	HP Operations Agent 11.xx	HPE Operations Agent 12.xx
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.	No admitido

Preguntas más frecuentes

¿Cómo verifico que hay datos antiguos después de una actualización?

Siga estos pasos:

1. Después de actualizar a HPE Operations Agent 12.xx, inicie sesión con los privilegios raíz.
2. Ejecute el comando siguiente:

   ```
   <OvInstallBinDir>ovcodautil -obj
   ```

3. Se genera una lista que consiste en todos los orígenes de datos, clases y métricas.

¿Cómo suprimo los datos antiguos después de una actualización?

Siga los pasos para quitar archivos de registro antiguos.

1. Inicie sesión en HPE Operations Agent 12.xx con los privilegios raíz.
2. Ejecute el comando siguiente para detener el proceso **oacore**:

   ```
   ovc -stop oacore
   ```
3. Suprima los archivos siguientes:

   En Windows:

   ```
   <OvDataDir>/conf/perf/datasources

   <OvDataDir>/datafiles/coda*

   <OvDataDir>/datafiles/log*
   ```

   En HP-UX/Linux/Solaris:

   ```
   <OvDataDir>/conf/perf/datasources

   <OvDataDir>/datafiles/coda*

   /var/opt/perf/datafiles/log*
   ```
4. Ejecute el comando siguiente para iniciar el proceso oacore:

   ```
   ovc -start oacore
   ```

Cómo volver a crear el Almacén de datos de métricas?

Siga los pasos para volver a crear el Almacén de datos de métricas:

1. Ejecute el comando siguiente para detener el proceso **oacore**:

   ```
   ovc -stop oacore
   ```
2. *Opcional:* Realice una copia de seguridad del directorio de base de datos existente `<OvDataDir>databases/oa/`
3. Ejecute el comando siguiente para eliminar el almacén de datos:

   En Windows:

   ```
   del /F "<OvDataDir>databases\oa\*"

   "<OvInstallBinDir>\sqlite3.exe" "%OvDataDir%databases\oa\oa.db"
   <"%OvDataDir%conf\oa\Model\DMLMetaMetaSchema"
   ```

   En HP-UX/Linux/Solaris/AIX:

   ```
   rm -f <OvDataDir>/databases/oa/*
   ```

```
<OvInstallBinDir>/sqlite3 /var/opt/OV/databases/oa/oa.db
<OvDataDir>/conf/oa/Model/DMLMetaMetaSchema
```

4. Ejecute el comando siguiente para establecer la variable de configuración UPDATED_MODEL_AVAILABLE en *True.*

   ovconfchg -ns oacore -set UPDATED_MODEL_AVAILABLE TRUE

5. Ejecute el comando siguiente para iniciar el proceso **oacore**:

   ```
   ovc -start oacore
   ```

Nota: Una clase predeterminada (usada a través de API o DSI de envío) se vuelve a crear automáticamente en un envío posterior.

Enviar información de la documentación

Si desea realizar comentarios sobre este documento, puede ponerse en contacto con el equipo de documentación por correo electrónico. Si hay un cliente de correo electrónico configurado en este sistema, haga clic en el vínculo correspondiente y se abrirá una ventana de correo electrónico con la siguiente información en la línea del asunto:

Comentarios sobre Guía de usuario (Operations Agent 12.01)

Escriba su comentario en el correo electrónico y haga clic en enviar.

Si no dispone de cliente de correo electrónico, copie la información anterior en un nuevo mensaje de cliente de correo web y envíe sus comentarios a docfeedback@hpe.com.

Agradecemos sus comentarios.