



Windows-Ereignisse mit Log Parser überwachen

von  Nils Kaczinski  14. November 2009, 22:12 Uhr

 Kategorie: Administration, Downloads, Log Parser, Scripting, Tools, Troubleshooting



The screenshot shows a web interface for Log Parser. It includes a title 'faq-o-matic.net Error Reporting: Server Event Logs', a 'Hidden Events' section, a table for 'Number of Errors and Warnings per Server in the Last 24 Hours', a table for 'Repeated Events of the Last Week', and sections for 'Non-Informational Events of the Last Hour' and 'Non-Informational Events of the Last 24 Hours'.

ComputerName	EventID	SourceName	Number of Events
DC01	Warning event		12
DC01	Error event		6
DC02	Success Audit event		139

EventID	SourceName	Number of Events
Application	215	18
Application	1019	88
Application	1019	88
Application	1019	216
Application	1011	4248
Directory Service	1004	9
Directory Service	1009	180
Directory Service	1002	180
Directory Service	1011	180

Microsofts kostenloser Log Parser ist ein höchst universelles Werkzeug. Mit seinen flexiblen Optionen kann man beispielsweise ein sehr leistungsfähiges, kostenloses Überwachungs- und Berichtssystem für Windows-Ereignisse auf einer größeren Zahl von Servern einrichten. Wir stellen hier eine Beispiel-Implementierung vor.

Das System stellt regelmäßig erzeugte Berichte über die Ereignisprotokolle ausgewählter Windows-Server als dynamische HTML-Dateien zur Verfügung, die einen schnellen Überblick über den Zustand der Server zulassen. Dabei konzentrieren sich die Berichte auf relevante Informationen, also z.B. Fehler- und Warnungs-Ereignisse oder solche Events, die in der letzten Zeit mehrfach aufgetreten sind. Der Hauptteil des Berichts listet alle nicht-informativen Ereignisse der Server in ihrer zeitlichen Abfolge auf. So kann man

Zusammenhänge zwischen Problemen erkennen, die auf mehr als einer Maschine bestehen oder sich entwickeln.

Aufbau

Das System besteht aus mehreren Komponenten:

1. Microsoft Log Parser – das vielseitige Werkzeug zur Auswertung nahezu beliebiger Protokoll- und Datendateien. Es steht kostenlos zur Verfügung (Download-Link siehe unten).
2. Eine Dateifreigabe auf einem beliebigen Windows-Rechner im Netzwerk (genauer gesagt: Auf einem beliebigen CIFS-Server).
3. Eine Reihe von Skriptdateien, die man vor Gebrauch ein wenig anpassen muss und nahezu beliebig erweitern kann.

Download Log Parser:

[Download details: Log Parser 2.2]

<http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=890cd06b-abf8-4c25-91b2-f8d975cf8c07>

Funktionsweise

Das Berichtssystem setzt voraus, dass auf allen zu überwachenden Servern Log Parser installiert ist (die Installation ist unkritisch und läuft nach unserer Kenntnis auf allen Windows-Systemen von NT bis 2008 R2, auch auf x64-Installationen). Über zwei geplante Tasks exportiert Log Parser auf diesen Systemen jede Stunde (oder in anderen Intervallen) die Events der letzten Woche (auch hier sind andere Intervalle möglich) in eine CSV-Datei und kopiert diese dann auf eine zentrale Freigabe.

Der Admin, der die Logs überprüfen möchte, ruft auf seiner Workstation (wo ebenfalls Log Parser installiert sein muss) ein Batch auf. Dieses kopiert die Exportdateien aller Server zu einer Datei zusammen und wertet diese dann nach verschiedenen Kriterien aus. Die Ergebnisse dieser Auswertung schreibt das Batch dann in eine HTML-Datei und kopiert diese ebenfalls in die Freigabe.

In unserer Implementierung zeigt der Report folgende Auswertungen an (siehe auch Screenshot oben):

- Die Anzahl der Warnungen und Fehler in den letzten 24 Stunden für jeden Server
- Ereignisse, die in der letzten Woche wiederholt aufgetreten sind
- Kritische Ereignisse der letzten Stunde, geordnet nach der Uhrzeit und den Servern (so stehen Ereignisse, die zur selben Zeit auf verschiedenen Servern auftraten, beieinander – dank Konvertierung in UTC-Zeit auch weltweit)
- Kritische Ereignisse der letzten 24 Stunden, geordnet nach der Uhrzeit und den Servern (so stehen Ereignisse, die zur selben Zeit auf verschiedenen Servern auftraten, beieinander)

Das System lässt sich prinzipiell beliebig um weitere Auswertungen erweitern.

Die Überwachung verzichtet bewusst auf die "letzte" Aktualität und wertet die Ereignisse nur einmal stündlich aus. Es handelt sich also nicht um eine Alarmierung, sondern um eine Art Gesundheitsprüfung der Umgebung. Sie ist aber gut geeignet, um Probleme der Server frühzeitig festzustellen und Zusammenhänge zu erkennen, vor allem wenn mehr als eine Maschine beteiligt ist.



The screenshot shows a web-based report interface. At the top, there's a title bar and a header section. Below that is a table with columns for 'Server', 'Event', 'Time', and 'Details'. The table contains several rows of data, with the 'Details' column providing a text-based description of each event. The interface has a classic Windows-style look with a yellowish background and blue headers.

Die zeitliche Beschränkung auf eine Auswertung pro Stunde ist sehr ressourcenschonend und eignet sich auch für große, weltweit verteilte Umgebungen. Ebenso ist die Architektur, dass jeder Server seine Ereignisse selbst exportiert und sie danach gesammelt auf eine Freigabe kopiert, für solche Umgebungen geeignet, denn auf diese Weise entsteht nur eine geringe Belastung des Netzwerks. Zudem funktioniert das gesamte Reporting auf diese Weise auch dann, wenn ein einzelner Server mal nicht erreichbar ist.

In sehr großen Umgebungen ist es oft sinnvoll, die Server in mehrere Gruppen zu unterteilen, für die man mit dem System separate Berichte erzeugt. Um das zu erreichen, benötigt man im Wesentlichen für jede Servergruppe einen eigenen Ordner in der Freigabe sowie leicht angepasste Batch-Dateien auf der Admin-Workstation und den Servern.

Download

Der Download findet sich hier:

 [Error Reporting mit Log Parser](#) (7.7 KiB, 33 hits)

Anpassung und Installation

Möchte man die vorliegende Beispiel-Implementierung übernehmen, so muss man nur folgende Dateien anpassen:

Ordner "Error Reporting"

- **MergeEvents.bat:** Anpassen der Pfade ganz am Anfang des Skripts (in den SET-Anweisungen). Im Wesentlichen nur in der Variablen *FOLDER* den freigegebenen Ordner eintragen, in dem die Logs und der Bericht gespeichert werden.
- **CreateErrorReport.bat:** Die Variable *ServerGroupName* nimmt einen beschreibenden Namen für die Servergruppe auf, deren Ereignisse ausgewertet werden. Außerdem auch hier Anpassen der Pfade ganz am Anfang des Skripts (in den SET-Anweisungen). In die Variable *TARGETFILE* den Pfad zur erzeugten HTML-Datei angeben, die in der Freigabe liegen sollte. In *LPPFILES* den Pfad zur Programmdatei von Log Parser eintragen (normalerweise *%ProgramFiles%\Logparser 2.2\logparser.exe*). In die Variable *LogFile* den Pfad zur Datei mit den gesammelten Events eintragen (normalerweise *AllEvents.txt* in der Freigabe).

Ordner "Server"

- **Install_Logparser-Tasks.bat:** Bei dieser Datei ist die Anpassung leider am aufwändigsten, weil sie jeweils von der Sprachversion des Windows abhängt, von der die Ereignisse gesammelt werden sollen. In der Variablen *LocalPath* den Pfad angeben, in dem die Skriptdateien später liegen sollen. In die Variable *AdminGroupName* den Namen der lokalen Administratoren-Gruppe eintragen (deutsches Windows: "Administratoren", englisches Windows: "Administrators" usw.). In die Variable *HourlyName* muss der Ausdruck, den *schtasks.exe* für eine stündliche Ausführung benötigt (englisches Windows: "hourly", deutsches Windows: "stündlich" [in der korrekten Codepage, also "stündlich"; am besten kopiert den korrekten Ausdruck aus der *rem*-Zeile] – wer ist auf die blöde Idee gekommen, das zu lokalisieren?!).
- **EventCollector.bat:** Die Variable *LPPFILES* enthält auch hier den Pfad zur Programmdatei von Log Parser (normalerweise *%ProgramFiles%\Logparser 2.2\logparser.exe*). *LocalPath* ist wieder der Pfad, in dem die Skriptdateien später liegen sollen.
- **EventCollectorCopy.bat:** *LocalPath* ist der Pfad, in dem die Skriptdateien später liegen sollen. In der Variablen *FOLDER* den freigegebenen Ordner eintragen, in dem die Logs und der Bericht gespeichert werden.
- **EventCollector.sql:** Hier in der Zeile, die mit *INTO* beginnt, den Pfad zur Ausgabedatei angeben. Dieser muss identisch sein mit dem Pfad, der in *EventCollectorCopy.bat* als *LocalPath* eingetragen ist.

Nach dieser Anpassung geht es an die Installation:

- Auf allen Servern, deren Ereignisse ausgewertet werden sollen, installiert man *Log Parser*. Das dauert nur ein paar Momente. Im Prinzip reicht es auch aus, die Datei *logparser.exe* aus einer bestehenden Installation auf den Server zu kopieren, aber das geht nur unwesentlich schneller.
- *Log Parser* installiert man ebenso auf allen Admin-Workstations, von denen aus dann die Reports erzeugt werden. Zum reinen Ansehen fertiger Berichte reicht natürlich ein Webbrowser.
- Den ganzen Ordner *Server* kopiert man auf jeden zu überwachenden Server. Dort ruft man einmal das oben angepasste Batch *Install_Logparser-Tasks.bat* auf. Es kopiert die Skriptdateien an den Zielort und erzeugt die nötigen Tasks, die die Überwachung ausführen. Dabei muss man einmal sein Kennwort eingeben, denn einer der beiden Tasks wird für das gerade angemeldete Konto erzeugt. Wer das ändern möchte, tut dies hinterher direkt in der Task-Verwaltung. Weiterhin setzt das Batch Berechtigungen für diesen User (Lesen) sowie für die *Administratoren*-Gruppe (Vollzugriff) auf die Batchdateien, die zu den Tasks gehören. Anderenfalls wären Angriffe möglich, wenn ein unbefugter User die Batchdateien manipuliert, denn der Server führt sie ja regelmäßig aus.
- Auf einem Server richtet man die Freigabe ein, in der die Daten gesammelt werden sollen. Der UNC-Pfad dieser Freigabe muss identisch sein mit dem, der oben in die Skriptdateien eingetragen wurde. Schreibrechte müssen für den User bestehen, der zur Ausführung der Tasks verwendet wird, sowie für alle Admins, die die Reports erzeugen sollen. Leserechte brauchen alle User, die die fertigen Reports nutzen sollen.
Idealerweise kopiert man auch die Datei *styles.css* in die Freigabe, damit der Report hinterher auch mit den richtigen Formaten angezeigt wird.

Das war es auch schon. Direkt nach dem Einrichten sollte man auf jedem Server die beiden Tasks einmal ausführen – erst "LogParserEventCollector" (erzeugt die Event-Sammlung unter dem Namen "eventcollector.txt" im lokalen Ordner; das kann durchaus ein paar Momente dauern), danach "LogParserEventCopy" (kopiert diese Datei mit vorangestelltem Servernamen in die Freigabe).

Nutzung

Das System erzeugt die Berichte standardmäßig nur auf Anforderung. Dazu ruft man von einer Admin-Workstation aus, auf der Log Parser installiert ist, das Batch *CreateErrorReport.bat* auf und wartet ein paar Momente (in den meisten Umgebungen dürften das mehrere Minuten sein – je nach Zahl der Server und der Events). Danach öffnet sich von selbst der Browser mit dem Bericht.

Da die Ereignis-Tabellen recht umfangreich sein können, sind sie im Bericht zunächst ausgeblendet. Über die entsprechenden Links kann man sie auf- und zuklappen.

Die Liste der aufgetretenen Events ist mit zwei Komfortfunktionen ausgestattet: Zum einen kann man durch Anklicken der Event-ID technische Hinweise und Lösungsvorschläge von der Community-Webseite *eventid.net* aufrufen. Zum anderen kann man die Übersichtlichkeit verbessern, indem man bestimmte Event-IDs aus dem Bericht ausblendet. Dazu klickt man auf "Hide ID". Alle ausgeblendeten IDs führt der Report am oberen Seitenrand auf, wo man sie per Klick wieder einblenden kann.

Erweiterung und Anpassung

Das gesamte System bedient sich im Wesentlichen der Möglichkeiten von Log Parser. So nutzt es HTML-Templates (*.tpl-Dateien), um die ausgelesenen Daten als HTML-Fragmente in die Reportdatei zu schreiben. Die Abfragen selbst sind in .sql-Dateien untergebracht, die man natürlich kopieren oder bearbeiten kann. Das Erzeugen der Reports kann man über den Taskplaner automatisieren, wobei es dann aber sinnvoll ist, das *start*-Kommando am Ende der Batchdatei *CreateErrorReport.bat* zu entfernen, das den Report im Browser aufruft.

Verwandte Beiträge:

1. [Auf Windows-Ereignisse reagieren](#)
Sehr oft möchte man als Administrator über bestimmte Ereignisse die im Ereignisprotokoll erscheinen, sofort informiert werden. Hier gibt es verschiedene...
2. [Drag & Drop ins CMD-Fenster geht wieder](#)
Mit Windows Vista hatte Microsoft einige Verhaltensweisen der Desktop-Oberfläche verändert. Eine kleine, aber für manchen gewichtige Änderung: Es war nicht...
3. [Windows-Ereignisse erzeugen](#)
Windows-Ereignisse im Ereignisprotokoll lassen sich per Skript erzeugen. Das kann hilfreich sein, wenn man die Funktion des Eventlogs prüfen möchte,...
4. [fcp: Dateien und Ordner kopieren](#)
fcp.vbs, Version: 1.0, ist ein Skript zum Kopieren von ganzen Ordner-Hierarchien, einzelnen Ordnern, Gruppen von Dateien und einzelnen Dateien. Es erlaubt...
5. [Schema-Versionen vergleichen](#)
In dem kürzlich hier erschienenen Artikel "Schema-Erweiterungen auffinden" habe ich einen manuellen Weg beschrieben, mit dem sich Erweiterungen im Schema...