

در [مبحث قبلی](#) گفتیم که ویرایش تنظیمات لاگ‌ها از طریق IIS با ویرایش مستقیم فایل‌های کانفیگ میسر است. در این مقاله که قسمت پایانی مبحث لاگ هاست، در مورد ویرایش فایل‌های کانفیگ صحبت می‌کنیم؛ همچنین استفاده از دستورات appcmd برای ویرایش و نهایتاً کد نویسی در زبان سی شارپ و جاوااسکریپت. تنظیمات لاگ سایت‌ها در فایل applicationhost در آدرس زیر قرار دارد:

C:\Windows\System32\inetsrv\config\applicationHost.config

برای هر تگ سایت، یک تگ <logfile> وجود دارد که ویژگی‌های Attributes آن، نوع ثبت لاگ را مشخص می‌کنند و می‌توانید مستقیماً در اینجا به ویرایش بپردازید. البته ویرایش فایل کانفیگ از طریق IIS به طور مستقیم هم امکان پذیر است. برای این منظور در IIS سرور را انتخاب و از بین ماژول‌های قسمت management گزینه‌ی Configuration Editor را انتخاب کنید. در قسمت Section گزینه‌ی System.applicationhost را باز کرده و از زیر مجموعه‌های آن گزینه‌ی Site را برگزینید. در تنظیمات باز شده، گزینه collection را انتخاب کنید تا در انتهای سطر، دکمه‌ی ... پدیدار گردد. روی آن کلیک کنید تا محیطی ویرایشی باز گردد که به شما اجازه‌ی افزودن و ویرایش خصوصیت‌ها را می‌دهد. برای ویرایش لاگ‌ها باید خصوصیت logfile را باز کنید. اگر قسمت قبلی را مطالعه کرده باشید، باید بسیاری از این خصوصیت‌ها و مقادیر را بشناسید. خصوصیات دیگری را هم مشاهده خواهید کرد که شاید قبلاً ندیده‌اید که البته بستگی به ورژن IIS شما دارد؛ مثلاً خصوصیت‌های flushByEntryCountW3Clog و maxLogLineLength از IIS8.5 اضافه شده‌اند.

#### جدول خصوصیت‌ها

خصوصیت	توضیح
customLogPluginClsid	یک پارامتر رشته‌ای اختیاری که در آن، آی دی کلاس یا کلاس‌هایی نوشته می‌شود که برای custom logging نوشته شده‌اند و این گزینه ترتیب اجرای آن‌ها را تعیین می‌کند.
directory	اختیاری است. محل ذخیره‌ی لاگ فایل‌ها را مشخص می‌کند و در صورتیکه ذکر نشود، همان مسیر پیش فرض است.
enabled	اختیاری است. فعال بودن سیستم لاگ برای آن سایت را مشخص می‌کند. مقدار پیش فرض آن true است.
flushByEntryCountW3Clog	این مقدار مشخص می‌کند چند رخداد باید اتفاق بیفتد تا عمل ذخیره سازی لاگ صورت گیرد. اگر بعد از هر رخداد عمل ثبت لاگ انجام شود، سرعت ثبت لاگ‌ها بالا می‌رود؛ ولی باعث استفاده‌ی مداوم از منابع و همچنین درخواست ثبت اطلاعات را روی دیسک خواهد داد و تاوان آن با زیاد شدن عملیات روی دیسک، پرداخته خواهد شد. ولی در حالتیکه چند رخداد را نگهداری سپس دسته‌ای ثبت کند، باعث افزایش کارایی و راندمان سرور خواهد شد. در صورتیکه سرور به مشکلات لحظه‌ای برخورد می‌کند مقدار آن را کاهش دهید. مقدار پیش فرض 0 است. یعنی اینکه ثبت، بعد از 64000 لاگ خواهد بود.
localTimeRollover	نحوه‌ی نامگذاری فایل‌های لاگ را مشخص می‌کند که مقدار بولین گرفته و اختیاری است. به طور پیش فرض مقدار false دارد.

خصوصیت	توضیح
<b>logExtFileFlags</b>	این گزینه در حالتی به کارتان می‌آید که فرمت W3C را برای ثبت لاگ‌ها انتخاب کرده باشید و در اینجا مشخص می‌کنید که چه فیلدهایی باید در لاگ باشند و اگر بیش از یکی بود میتوان با ، (کاما) از هم جدایشان کرد.
<b>logFormat</b>	نوع فرمت ذخیره سازی لاگ‌ها
<b>logSiteId</b>	اختیاری است و مقدار پیش فرض آن true است. بدین معنا که کد یا شماره‌ی سایت هم در لاگ خواهد بود و این در حالتی است که گزارش در سطح سرور باشد. در غیر این صورت اگر هر سایت، جداگانه لاگی برای خود داشته باشد، ذکر نمی‌گردد.
<b>logTargetW3C</b>	اختیاری است و مقدار file و * ETW را می‌گیرد که به طور پیش فرض روی File تنظیم است. در این حالت فایل لاگ‌ها در یک فایل متنی توسط http.sys ذخیره می‌شود. ولی موقعیکه از ETW استفاده می‌شود، http.sys با استفاده از iislogprovider داده‌ها را به سمت ETW ارسال میکند که منجر به اجرای سرویس Logsvc شده که از داده‌ها کوئری گرفته و آن‌ها را مستقیماً از پروسه‌های کارگر جمع‌آوری و به سمت فایل لاگ ارسال می‌کند. همچنین انتخاب این دو گزینه نیز ممکن است.
<b>maxLogLineLength</b>	حداکثر تعداد خطی که یک لاگ میتواند داشته باشد تا اینکه بتوانید در مصرف دیسک سخت صرفه جویی کنید و بیشتر کاربرد آن برای لاگ‌های کاستوم است. این عدد باید از نوع Uint باشد و اختیاری است و از 2 تا 65536 مقدار میپذیرد که مقدار پیش فرض آن 65536 می‌باشد.
<b>period</b>	همان مبحث زمان بندی در مورد ایجاد فایل‌های لاگ است که در مقاله‌ی پیشین بررسی کردیم و مقادیر Dially, Hourly, monthly و weekly را می‌پذیرد. همچنین maxsize هم هست؛ موقعی که لاگ به نهایت حجمی که برای آن تعیین کردیم میرسد.
<b>truncateSize</b>	اختیاری است و مقدار آن از نوع int64 است. حداکثر حجم یک فایل لاگ را مشخص می‌کند تا اگر period روی maxsize تنظیم شده بود، حداکثر حجم را میتوان از اینجا تعیین نمود. در مقاله پیشین در این باره صحبت کردیم؛ حداقل عدد برای آن 1,048,576 است و اگر کمتر از آن بنویسید، سیستم همین عدد 1,048,576 را در نظر خواهد گرفت. مقدار پیش فرض آن 20971520 می‌باشد.

\* ETW یا [Event Tracing Windows](#) ، سیستم و یا نرم افزاری برای عیب یابی و نظارت برای کامپوننت‌های ویندوزی است و یکی از استفاده کننده‌هایش IIS است که از ویندوز 2000 به بعد اضافه شده‌است. برای قطع کردن این ماژول در IIS هم میتوانید [قسمت هفتم](#) را بررسی نمایید و دنیال ماژول TracingModule بگردید. این ماژول به صورت Real time به ثبت رخدادهای IIS می‌پردازد.

به غیر از خصوصیات بالا، خصوصیت customFields نیز از IIS 8.5 (به بعد) در دسترس است. اگر قصد دارید به غیر از فیلدهای W3C فیلدهای اختصاصی دیگری نیز داشته باشید، میتوان از این گزینه استفاده کرد. این فیلدهای کاستوم می‌توانند اطلاعاتشان را از request header ، response header و server variables دریافت کنند. این ویژگی تنها در فرمت W3C و در سطح سایت قابل انجام است. موقعی که یک فایل لاگ شامل فیلدهای اختصاصی شود، به انتها نام فایل \_X اضافه میگردد تا نشان دهد شامل یک فیلد

اختصاصی یا کاستوم است. نحوه تعریف آن در فایل applicationhost به شکل زیر است:

```
<system.applicationHost>
  <sites>
    <siteDefaults>
      <logFile logFormat="W3C"
        directory="%SystemDrive%\inetpub\logs\LogFiles"
        enabled="true">
      <customFields>
        <clear/>
        <add logFieldName="ContosoField" sourceName="ContosoSource"
          sourceType="ServerVariable" />
      </customFields>
    </logFile>
  </siteDefaults>
</sites>
</system.applicationHost>
```

### تغییر تنظیمات لاگ با Appcmd

```
appcmd.exe set config -section:system.applicationHost/sites /siteDefaults.logFile.enabled:"True"
/commit:apphost
appcmd.exe set config -section:system.applicationHost/sites /siteDefaults.logFile.logFormat:"W3C"
/commit:apphost
appcmd.exe set config -section:system.applicationHost/sites
/siteDefaults.logFile.directory:"%SystemDrive%\inetpub\logs\LogFiles" /commit:apphost
```

تنظیمات تگ لاگ با برنامه نویسی و اسکریپت نویسی همچنین با رفرنس Microsoft.web.administration در پروژه‌های دات نتی خود می‌توانید امکان ویرایش تنظیمات را در برنامه‌های خود نیز داشته باشید:

```
using System;
using System.Text;
using Microsoft.Web.Administration;

internal static class Sample
{
    private static void Main()
    {
        using (ServerManager serverManager = new ServerManager())
        {
            Configuration config = serverManager.GetApplicationHostConfiguration();
            ConfigurationSection sitesSection = config.GetSection("system.applicationHost/sites");
            ConfigurationElement siteDefaultsElement = sitesSection.GetChildElement("siteDefaults");

            ConfigurationElement logFileElement = siteDefaultsElement.GetChildElement("logFile");
            logFileElement["logFormat"] = @"W3C";
            logFileElement["directory"] = @"%SystemDrive%\inetpub\logs\LogFiles";
            logFileElement["enabled"] = true;

            serverManager.CommitChanges();
        }
    }
}
```

با استفاده از اسکریپت نویسی توسط جاوا اسکریپت و وی بی اسکریپت هم نیز این امکان مهیاست:

```
var adminManager = new ActiveXObject('Microsoft.ApplicationHost.WritableAdminManager');
adminManager.CommitPath = "MACHINE/WEBROOT/APPHOST";
var sitesSection = adminManager.GetAdminSection("system.applicationHost/sites",
"MACHINE/WEBROOT/APPHOST");
var siteDefaultsElement = sitesSection.ChildElements.Item("siteDefaults");

var logFileElement = siteDefaultsElement.ChildElements.Item("logFile");
logFileElement.Properties.Item("logFormat").Value = "W3C";
logFileElement.Properties.Item("directory").Value = "%SystemDrive%\\inetpub\\logs\\LogFiles";
logFileElement.Properties.Item("enabled").Value = true;

adminManager.CommitChanges();
```

## FTP Logging

برای اطمینان از نصب Ftp logging موقع نصب، باید از مورد زیر مطمئن باشید:

IIS را باز کنید و در لیست درختی، سرور را انتخاب کنید. در قسمت FTP میتوانید گزینهی Ftp logging را ببینید. تنظیمات این قسمت هم دقیقا همانند قسمت logging میباشد و همان موارد برای آن هم صدق میکند.

## بررسی تگ آن در applicationhost

تگ این نوع لاگ در فایل applicationhost در زیر مجموعهی تگ <site> به شکل زیر نوشته میشود:

```
<system.ftpServer>
  <log centralLogFileMode="Central">
    <centralLogFile enabled="true" />
  </log>
</system.ftpServer>
```

گزینه **centralLogFileMode** دو مقدار **central** و **site** را میپذیرد. اگر گزینهی **central** انتخاب شود، یعنی همهی لاگها را داخل یک فایل در سطح سرور ثبت کن ولی اگر گزینهی **site** انتخاب شده باشد، لاگ هر سایت در یک فایل ثبت خواهد شد.

گزینهی **logInUTF8** یک خصوصیت اختیاری است که مقدار پیش فرض آن **true** میباشد. در این حالت باید تمامی رشتهها به انکدینگ UTF-8 تبدیل شوند.

همانطور که میبینید تگ **log** در بالا یک تگ فرزند هم به اسم **centralLogFile** دارد که همان خصوصیات جدول بالا در آن مهبیاست.

## دسترسی به تنظیمات این قسمت توسط دستور Appcmd:

```
appcmd.exe set config -section:system.ftpServer/log /centralLogFileMode:"Central" /commit:apphost
appcmd.exe set config -section:system.ftpServer/log /centralLogFile.enabled:"True" /commit:apphost
```

## دسترسی به تنظیمات این قسمت توسط دات نت:

```
using System;
using System.Text;
using Microsoft.Web.Administration;

internal static class Sample
{
    private static void Main()
    {
        using (ServerManager serverManager = new ServerManager())
        {
            Configuration config = serverManager.GetApplicationHostConfiguration();

            ConfigurationSection logSection = config.GetSection("system.ftpServer/log");
            logSection["centralLogFileMode"] = @"Central";

            ConfigurationElement centralLogFileElement = logSection.GetChildElement("centralLogFile");
            centralLogFileElement["enabled"] = true;

            serverManager.CommitChanges();
        }
    }
}
```

### دسترسی به تنظیمات این قسمت توسط Javascript:

```
var adminManager = new ActiveXObject('Microsoft.ApplicationHost.WritableAdminManager');
adminManager.CommitPath = "MACHINE/WEBROOT/APPHOST";

var logSection = adminManager.GetAdminSection("system.ftpServer/log", "MACHINE/WEBROOT/APPHOST");
logSection.Properties.Item("centralLogFileMode").Value = "Central";

var centralLogFileElement = logSection.ChildElements.Item("centralLogFile");
centralLogFileElement.Properties.Item("enabled").Value = true;

adminManager.CommitChanges();
```