

پیکربندی قسمت لاگ‌ها، میتواند برای یک سرور و یا وب سایت خاص از طریق فایل کانفیگ یا از طریق خود IIS انجام گیرد. برای اینکه به بیشتر این قابلیت‌ها در IIS دسترسی داشت، باید یکی از نسخه‌های ویندوز سرور 2012 و ویندوز 8 را نصب کرده باشید. لاگ‌ها به ثبت خطاها و درخواست‌های HTTP می‌پردازند و با تحلیل آن‌ها میتوان عملیات بهینه سازی را بر روی سرو اجرا کرد. تمامی ثبت لاگ‌ها توسط Http.sys انجام می‌گیرد.

### نحوه‌ی ذخیره سازی لاگ‌ها

در این بخش نحوه‌ی ذخیره سازی و فرمت ذخیره‌ی لاگ‌ها را در دو سطح سایت و سرور به طور جداگانه بررسی می‌کنیم. در IIS ماژول Logging را باز کنید و در لیست One log file per می‌توانید مشخص کنید که لاگ‌ها در چه سطحی اجرا شوند. اگر گزینه‌ی server باشد، تمامی خطاها و درخواست‌های رسیده به سرور در یک فایل لاگ ثبت می‌شوند. ولی اگر سطح سایت باشد، برای هر سایت بر روی IIS لاگ‌ها، جداگانه بررسی می‌شوند. به طور پیش فرض سطح سایت انتخاب شده است.

### سطح سایت

موقعی که در لیست، سایت را انتخاب کنید، در لیست format می‌توانید تعیین کنید که لاگ‌ها به چه صورتی باید ذخیره شوند. مواردی که در این حالت لیست می‌شوند گزینه‌های W3C, IIS, NCSA, Custom می‌باشند که در زیر یکایک آن‌ها را بررسی می‌کنیم:

**فرمت IIS:** این فرمت توسط مایکروسافت ارائه شده و در این حالت لاگ‌های همه‌ی وب سایت‌ها ذخیره می‌شوند. به این فرمت Fixed ASCII Based Text نیز می‌گویند؛ چرا که اجازه‌ی خصوصی سازی ندارد و نمی‌توانید بگویید چه فیلدهایی در لاگ قرار داشته باشند. لاگ فایل‌های این فرمت با ، (کاما) از هم جدا می‌شوند و مقدار زمانی که برای هر فیلد ثبت می‌شود، به صورت محلی local Time می‌باشد.

فیلدهایی که در لاگ این نوع فرمت خواهند آمد، به شرح زیر است:

Client IP address

User name

Date

Time

Service and instance

Server name

Server IP address

Time taken

Client bytes sent

Server bytes sent

Service status code (A value of 200 indicates that the request was fulfilled successfully)

.Windows status code (A value of 0 indicates that the request was fulfilled successfully)

Request type

Target of operation

Parameters (the parameters that are passed to a script)

احتمال این وجود دارد که بعضی از فیلدها در بعضی رکوردها، شامل اطلاعاتی نباشند که به جای مقدار آن علامت - ثبت می‌گردد و برای کاراکترهایی که قابل نمایش نیستند یا کاراکتر نمایشی ندارند، از علامت + استفاده می‌شود. دلیل اینکار هم این است که ممکن است یک کاربر مهاجم، به ارسال اطلاعات کلیدهای کنترلی چون [Carriage return](#) اختصارا CR یا [Line Feed](#) به اختصار LF کند، که باعث شکسته شدن خط لاگ فایل می‌شود و در نتیجه از استاندارد خارج خواهد شد و هنگام خواندن آن هم با خطا روبرو می‌شویم؛ در نتیجه با جایگزینی چنین کاراکترهایی با + از این اتفاق جلوگیری می‌شود. شکل زیر نمونه ای از یک خط لاگ در این فرمت است:

```
192.168.114.201, -, 03/20/01, 7:55:20, W3SVC2, SERVER, 172.21.13.45, 4502, 163, 3223, 200, 0, GET, /DeptLogo.gif, -,
```

نام فیلد	نوع حالت مقداردهی	توضیح اتفاقات افتاده
Client IP address	192.168.114.201	آی پی کلاینت
User name	-	کاربر ناشناس است
Date	03/20/01	تاریخ فعالیت
Time	7:55:20	ساعت فعالیت
	Service and instance	W3SVC2
لاگی که مربوط به سایت خاصی می‌شود به صورت W3SVC# نمایش داده می‌شود که علامت # شماره سایت می‌باشد که در اینجا این لاگ مربوط به سایت شماره 2 است	Server name	SERVER
نام سرور	Server IP	172.21.13.45
آی پی سرور	Time taken	4502
چقدر انجام عملیات این درخواست به طول انجامیده است که بر حسب میلی ثانیه است.	Client bytes sent	163
تعداد بایت هایی که از طرف کلاینت به سرور ارسال شده است	Server bytes sent	3223
تعداد بایت هایی که از طرف سرور به سمت کلاینت ارسال شده است	Service status code	200
درخواست کاملاً موفقیت آمیز بوده است	Windows status code	0
درخواست کاملاً موفقیت آمیز بوده است	Request type	GET
نوع درخواست کاربر	Target of operation	/DeptLogo.gif
کاربر قصد دانلود یک فایل تصویری GIF	Parameters	-

نام فیلد	نوع حالت مقداردهی	توضیح اتفاقات افتاده
داشته است که نامش Deptlogo است		

**فرمت NCSA:** این فرمت توسط مرکز علمی کاربردهای ابرمحاسباتی [National Center for Supercomputing Applications](http://www.nsl.gov/research/projects/ncsa/) ایجاد شده و دقیقاً مانند قبلی نمیتوان در آن نوع فیلدها را مشخص کرد و برای جدا سازی، از فاصله space استفاده می‌کند و ثبت مقدار زمان در آن هم به صورت محلی و هم UTC می‌باشد.

این فیلدها در لاگ آن نمایش داده می‌شوند:

Remote host address

Remote log name (This value is always a hyphen)

User name

Date, time, and Greenwich mean time (GMT) offset

Request and protocol version

Service status code (A value of 200 indicates that the request was fulfilled successfully)

Bytes sent

نمونه ای از یک لاگ ثبت شده:

```
172.21.13.45 - Microsoft\JohnDoe [08/Apr/2001:17:39:04 -0800] "GET /scripts/iisadmin/ism.dll?http/serv
HTTP/1.0" 200 3401
```

نام فیلد	مقدار ثبت شده	توضیح اتفاق افتاده
Remote host address	172.21.13.45	آی پی کلاینت
Remote log name	-	نامی وجود ندارد
User name	Microsoft\JohnDoe	نام کاربری
Date, time, and GMT offset	[08/Apr/2001:17:39:04 -0800]	تاریخ و ساعت فعالیت به صورت محلی که 8 ساعت از مبدا گرینویچ بیشتر است
Request and protocol version	GET /scripts/iisadmin/ism.dll?http/serv HTTP/1.0	کاربر با متد GET و Http نسخه‌ی یک، درخواست فایل ism.dll را کرده است.
Service status code	200	عملیات کاملاً موفقیت آمیز بود.
Bytes sent	3401	تعداد بایت‌های ارسال شده به سمت کاربر

امنیت در برابر کاربران مهاجم مانند همان فرمت قبلی صورت گرفته است.

**فرمت W3C:** توسط W3C توسط کنسرسیوم جهانی وب ارائه شده است و یک فرمت customizable ASCII text-based است. به این معنی که میتوان فیلدهایی که در گزارش نهایی می‌آید را خودتان مشخص کنید، که برای اینکار در کنار لیست، دکمه‌ی Select

وجود دارد که میتوانید هر کدام از فیلدهایی را که خواستید، انتخاب کنید تا به ترتیب در خط لاگ ظاهر شوند. تاریخ ثبت به صورت UTC است.

نام فیلد	توضیح	به طور پیش فرض انتخاب شده است
Date	تاریخ رخ دادن فعالیت	بله
Time	ساعت رخ دادن فعالیت بر اساس UTC	بله
Client IP Address s	آی پی کلاینت	بله
User Name	نام کاربری که هویت آن تایید شده و در صورتی که هویت تایید شده نباشد و کاربر ناشناس باشد، جای آن - قرار می گیرد	بله
Service Name and Instance Number	نام و شماره سایتی که درخواست در آن صورت گرفته است	خیر
Server Name	نام سروری که لاگ روی آن ثبت می شود	خیر
Server IP Address	آی پی سرور که لاگ روی آن ثبت می شود	بله
Server Port	شماره پورتی که سرویس مورد نظر روی آن پورت اعمال می شود.	بله
Method	متد درخواست مثل GET	بله
URI Stem	هدف درخواست یا Target مثل index.htm	بله
URI Query	کوئری ارسال شده برای صفحات داینامیک	بله
HTTP Status	کد وضعیتی HTTP status	بله
Win32 Status	کد وضعیتی ویندوز	خیر
Bytes Sent	تعداد بایت های ارسال شده به سمت کلاینت	خیر
Bytes Received	تعداد بایت های دریافت شده از سمت کلاینت	خیر
Time Taken	زمان به طول انجامیدن درخواست بر حسب میلی ثانیه	خیر
Protocol Version	درخواست با چه نسخه ای از پروتکل http یا ftp ارسال شده است	خیر
Host	اگر در هدر درخواست ارسالی این گزینه بوده باشد، نوشته خواهد شد.	خیر
User Agent	اطلاعات را از هدر درخواست می گیرد.	بله
Cookie	اگر کوکی رد و بدل شده باشد، محتویات کوکی ارسالی یا دریافت شده	خیر
Referrer	کاربر از چه سایتی به سمت سایت ما آمده است.	خیر
Protocol Substatus		بله

نام فیلد	توضیح	به طور پیش فرض انتخاب شده است
	<p>در صورت رخ دادن خطا در IIS ، کد خطا بازگردانده میشود. در IIS به منظور امنیت بیشتر و کاهش حملات، محتوای خطاهای رخ داده در IIS به صورت متنی نمایش داده نمی‌شوند و شامل کد خطایی به اسم Substatus Code هستند تا مدیران شبکه با ردیابی لاگ‌ها پی به دلیل خطا و درخواست‌های ناموفق ببرند. برای مثال Error 404.2 به این معنی است که فایل درخواستی به دلیل قوانین محدود کننده، قفل شده و قابل ارائه نیست. ولی هکر تنها با خطای 404 یعنی وجود نداشتن فایل روبرو می‌شود. در حالت substatus code، کد شماره 2 را هم خواهید داشت که در لاگ ثبت می‌شود.</p> <p>هر شخصی که در سرور توانایی دسترسی به لاگ‌ها را داشته باشد، می‌تواند کد دوم خطا را نیز مشاهده کند. برای مثال مدیر سرور متوجه میشود که یکی از فایل‌های مورد نظر به کاربران، خطای 404 نمایش میدهد و با بررسی لاگ‌ها متوجه می‌شود که کد خطا 404.9 هست. از آنجا که ما همه‌ی کدها را حفظ نیستیم به این <a href="#">صفحه</a> رجوع می‌کنیم و متوجه میشویم تعداد کاربرانی که برای این فایل، اتصال connection ایجاد کرده‌اند بیش از مقدار مجاز است و مدیر میتواند این وضع را کنترل کند. برای مثال تعداد اتصالات مجاز را نامحدود unlimited تعیین کند.</p>	

حروف - و + برای موارد بالا هم صدق می‌کند. در ضمن گزینه‌های زیر در حالتی که درخواست از پروتکل FTP باشد مقداری نخواهند گرفت:

uri-query

host

(User-Agent)

Cookie

Referrer

substatus

**گزینه Custom :** موقعی که شما این گزینه را انتخاب کنید مازول logging غیرفعال خواهد شد. زیرا این امکان در IIS قابل پیکر بندی نیست و نوشتن مازول آن بر عهده شما خواهد بود؛ با استفاده از اینترفیس های ILogPluginEx ، ILogPlugin و ILogUIPlugin آن را پیاده سازی کنید.

#### ذخیره اطلاعات به انکدینگ UTF-8 و موضوع امنیت

در صورتی که شما از سایتی با زبانی غیر از انگلیسی و لاتین و فراتر از ANSI استفاده می کنید، این گزینه حتما باید انتخاب شده باشد تا درخواست را بهتر لاگ کند. حتی برای وب سایت های انگلیسی زبان هم انتخاب این گزینه بسیار خوب است؛ چرا که اگر به سمت سرور کاراکترهای خاصی در URL ارسال شوند، نمی تواند با کدپیچ موجود آن ها را درست تبدیل کند.

#### ادامه ی تنظیمات

موارد بعدی که در تنظیمات لاگ ها کاملا مشخص و واضح است، عملیات زمان بندی است که برای ساخت یک فایل لاگ جدید به کار می رود؛ برای مثال هر ساعت یک لاگ فایل جدید بسازد و فعالیت های موجود در هر ساعت در یک لاگ ذخیره می شوند. گزینه ی بعدی حداکثر حجم هر فایل لاگ است که به صورت بایت مشخص می شود. اگر مقداری که تعیین میکنید کمتر از 1048576 بایت باشد، خودش به طور پیش فرض همان 1048576 بایت را در نظر خواهد گرفت. گزینه بعدی do not create a new logfile بدین معناست که همه ی لاگ ها در یک فایل ذخیره می شوند و فایل جدیدی برای لاگ ها ایجاد نمی شود.

گزینه آخری به اسم use local time for filenames and rollover است که اگر انتخاب شود، نامگذاری هر فایل لاگ بر اساس زمان محلی ساخت فایل لاگ خواهد بود. در صورتیکه انتخاب نشود، نامگذاری با زمان UTC درج خواهد شد.

#### سطح سرور

لاگ ها فقط در سمت سرور انجام می گیرد و لاگ هر سایت در یک فایل لاگ ثبت می شود. اگر بخواهید لاگ ها را در سطح سرور انجام دهید، گزینه ی binary هم اضافه خواهد شد.

**Binary:** در این گزینه دیگر از قالب بندی یا فرمت بندی لاگ ها خبری نیست و لاگ هر وب سایت به صورت اختصاصی صورت نمی گیرد. عملیات ذخیره سازی و ثبت هر لاگ می تواند از منابع یک سرور از قبیل حافظه و CPU و ... استفاده کند و اگر تعداد این وب سایت ها بالا باشد، باقی روش ها باعث فشار به سرور می شوند. برای همین ایجاد یک فایل خام از لاگ ها در این مواقع می تواند راهگشا باشد. برای همه یک فایل لاگ ایجاد شده و بدون قالب بندی ذخیره می کند. پسوند این نوع لاگ ها ibl است که مخفف Internet Binary Log می باشد. دلیل این تغییر پسوند این است که اطمینان کسب شود کاربر، با برنامه های متنی چون notepad یا امثال آن که به Text Utilities معروفند فایل را باز نمی کند. برای خواندن این فایل های میتوان از برنامه ی [Log parser](#) استفاده کرد. پروتکل های FTP, NNTP و SMTP در این حالت لاگشان ثبت نمی شود.

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 شده اند.

#### جدول خصوصیت ها

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();
```