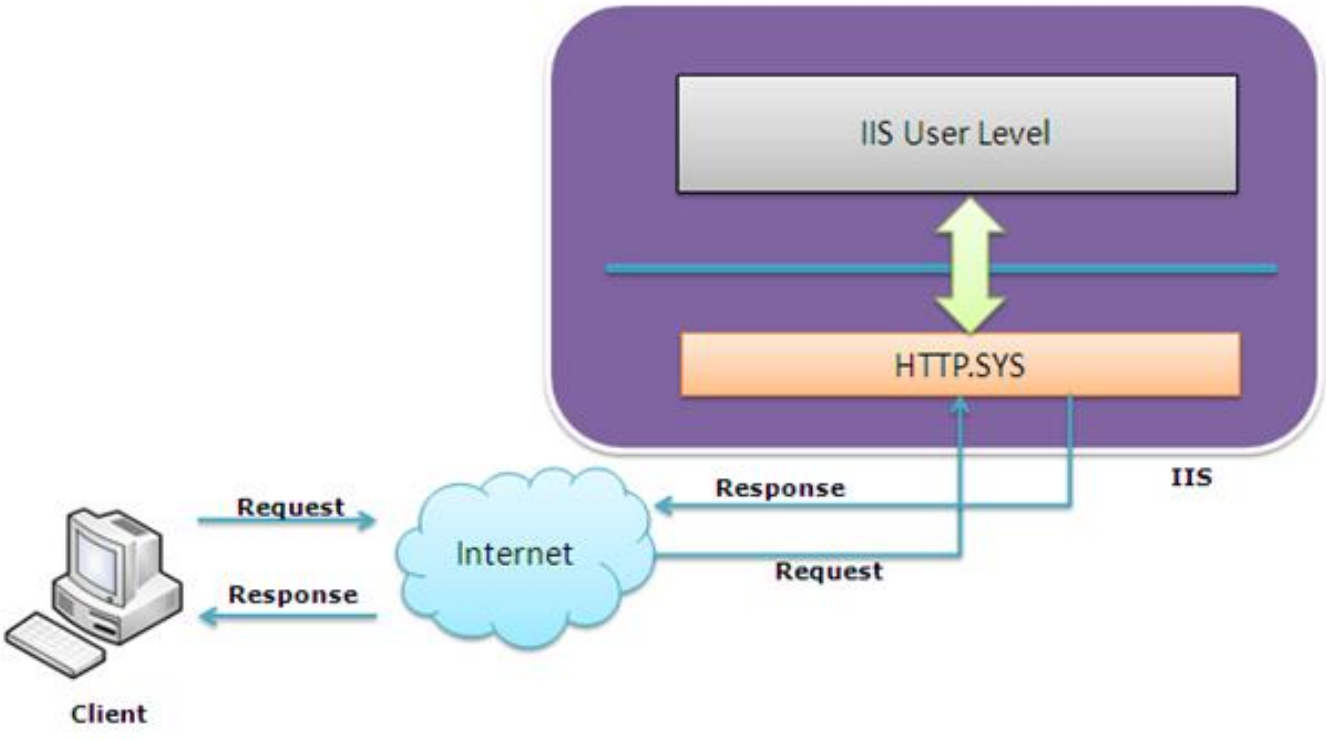
عنوان: **آشنایی با ساختار IIS قسمت اول**
نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۳/۱۰/۰۱ ۱:۲۰
آدرس: www.dotnettips.info
گروه‌ها: IIS, Web Capacity Analysis Tool -WCAT

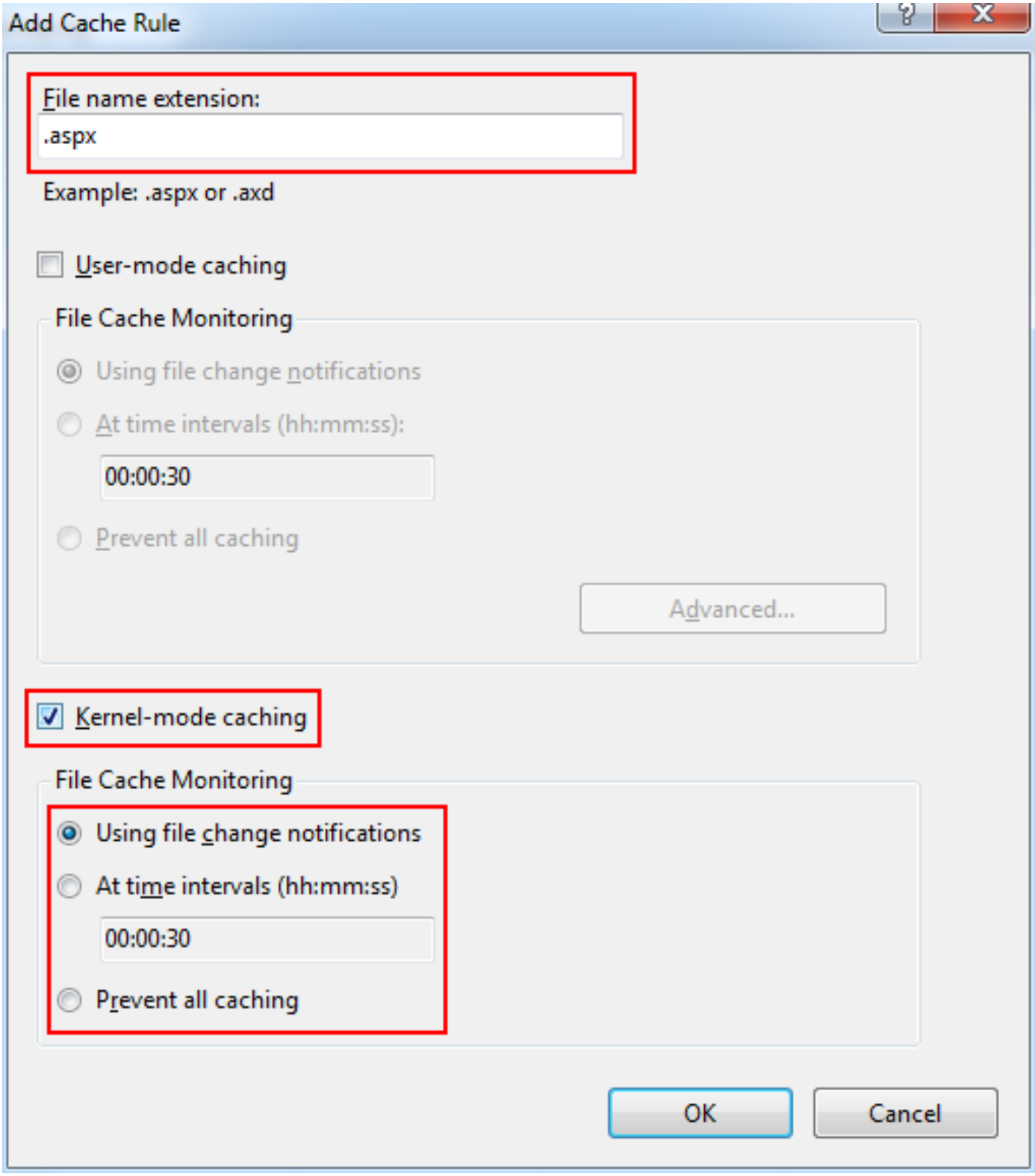
در مقاله قبل در مورد نحوه ذخیره سازی در حافظه نوشتیم و به user mode و kernel mode اشاراتی کردیم که می‌توانید به آن رجوع کنید.
در این سری مقالات قصد داریم به بررسی اجزا و روند کاری موجود در IIS بپردازیم که چگونه IIS کار می‌کند و شامل چه بخش هایی می‌شود. مطمئنا آشنایی با این بخش‌ها در روند شناسایی رفتارهای وب اپلیکیشن‌ها و واکنش‌های سرور، کمک زیادی به ما خواهد کرد. در اینجا نسخه IIS7 را به عنوان مرجع در نظر گرفته‌ایم.
وب سرور IIS در عبارت مخفف Internet information services به معنی سرویس‌های اطلاعاتی اینترنت می‌باشد. IIS شامل کامپوننت‌های زیادی است که هر کدام ازآن‌ها کار خاصی را انجام میدهند؛ برای مثال گوش دادن به درخواست‌های ارسال شده به سرور، مدیریت فرآیندها Process و خواندن فایل‌های پیکربندی Configuration؛ این اجزا شامل Http.sys، protocol listener و WSA و .. می‌شوند. **Protocol Listeners**
این پروتکل‌ها به درخواست‌های رسیده گوش کرده و آن‌ها را مورد پردازش قرار می‌دهند و پاسخی را به درخواست کننده، ارسال می‌کنند. هر listener بر اساس نوع پروتکل متفاوت هست. به عنوان مثال کلاینتی، درخواست صفحه‌ای را می‌کند و http listener که به آن Http.sys می‌گویند به آن پاسخ می‌دهد. به طور پیش فرض http.sys به درخواست‌های http و https گوش فرا می‌دهد، این کامپوننت از IIS6 اضافه شده است ولی در نسخه 7 از SSL نیز پشتیبانی می‌کند. **Http.sys یا Hypertext transfer protocol stack**
کار این واحد در سه مرحله دریافت درخواست، ارسال آن به واحد پردازش IIS و ارسال پاسخ به کلاینت است؛ قبل از نسخه 6 از Winsock یا windows socket api که یک کامپوننت user-mod بود استفاده می‌شد ولی Http.sys یک کامپوننت Kernel-mod هست.



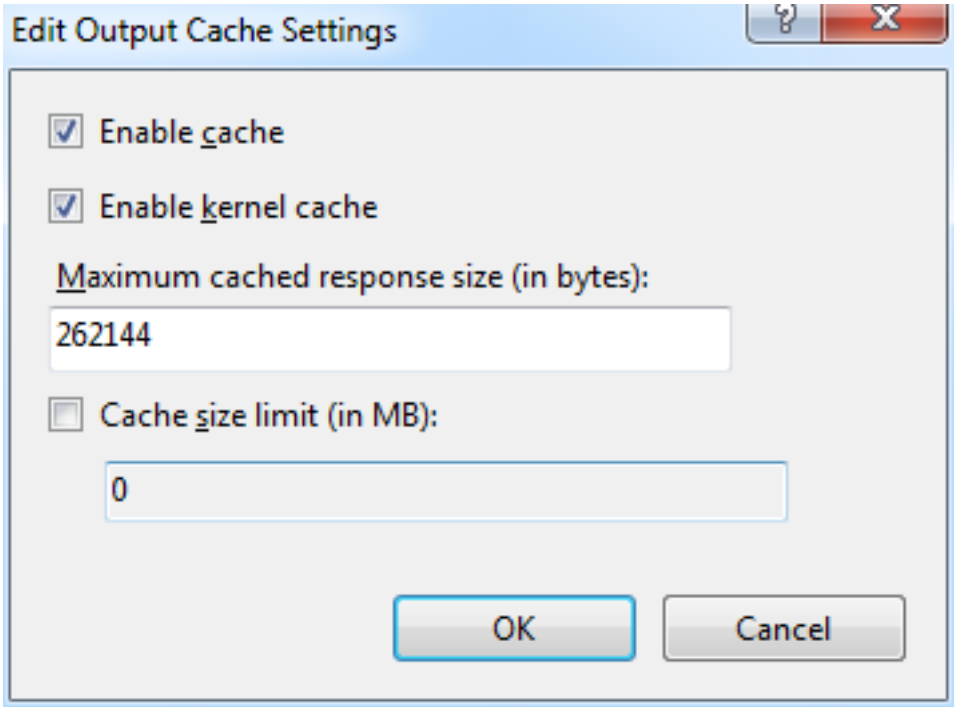
Http.sys مزایای زیر را به همراه دارد:

صف درخواست مد کرنل: به خاطر اینکه کرنل مستقیما درخواست‌ها را به پروسه‌های مربوطه میفرستد و اگر پروسه موجود نباشد، درخواست را در صف گذاشته تا بعدا پروسه مورد نظر آن را از صف بیرون بکشد.
برای درخواست‌ها یک پیش پردازش و همچنین اعمال فیلترهای امنیتی اعمال می‌گردد.
عملیات کش کردن تماما در محیط کرنل مد صورت می‌گیرد؛ بدون اینکه به حالت یوزرمد سوییچ کند. مد کرنل دسترسی بسیار راحت و مستقیمی را برای استفاده از منابع دارد و لازم نیست مانند مد کاربر به لایه‌های زیرین، درخواست کاری را بدهد؛ چرا که خود مستقیما وارد عمل می‌شود و برداشته شدن واسط در سر راه، موجب افزایش عمل caching می‌شود. همچنین دسترسی به کش باعث می‌شود که مستقیما پاسخ از کش به کاربر برسد و توابع پردازشی در حافظه بارگذاری نشوند. البته این کش کردن محدودیت هایی را هم به همراه دارد:
کش کرنل به صورت پیش فرض بر روی صفحات ایستا فعال شده است؛ نه برای صفحاتی با محتوای پویا که البته این مورد قابل تغییر است که نحوه این تغییر را پایینتر توضیح خواهیم داد.
اگر آدرس درخواستی شامل کوئری باشد صفحه کش نخواهد شد: *http://www.site.info/postarchive.htm?id=25*
برای پاسخ ازمکانیزم‌های فشرده سازی پویا استفاده شده باشد مثل gzip کش نخواهد شد
صفحه درخواست شده صفحه اصلی سایت باشد کش نخواهد شد : *http://www.dotnettip.info* ولی اگر درخواست بدین صورت باشه *http://www.domain.com/default.htm* کش خواهد کرد.
درخواست به صورت ناشناس anonymous نباشد و نیاز به authentication داشته باشد کش نخواهد شد (یعنی در هدر شامل گزینه authorization می‌باشد).
درخواست باید از نوع نسخه http1 به بعد باشد.
اگر درخواست شامل Entity-body باشد کش نخواهد کرد.
درخواست شامل If-Range/Range header باشد کش نمی‌شود.
کل حجم response بیشتر از اندازه تعیین شده باشد کش نخواهد گردید، این اندازه در کلید ریجستری UriMaxUriBytes قرار دارد. اطلاعات بیشتر
اندازه هدر بیشتر از اندازه تعیین شده باشد که عموما اندازه تعیین شده یک کیلو بایت است.
کش پر باشد، کش انجام نخواهد گرفت.

برای فعال سازی کش کرنل راهنمای زیر را دنبال کنید:
گزینه output cache را در IIS، فعال کنید و سپس گزینه Add را بزنید. کادر add cache rule که باز شود، از شما میخواهد یکی از دو نوع کش مد کاربر و مد کرنل را انتخاب کنید و مشخص کنید چه نوع فایل‌هایی (مثلا aspx) از این قوانین پیروری کنند و مکانیزم کش کردن به سه روش جلوگیری از کش کردن، کش زمان دار و کش بر اساس آخرین تغییر فایل انجام گردد.



برای تعیین مقدار سایز کش response که در بالا اشاره کردیم می‌توانید در همان پنجره، گزینه edit feature settings را انتخاب کنید.



این قسمت از مطلب که به نقل از مقاله آقای Karol Jarkovsky در این آدرس است یک سری تست هایی با نرم افزار(Web Capacity Analysis Tool (WCAT گرفته است که به نتایج زیر دست پیدا کرده است:

Kernel Cache Disabled 4 clients/160 threads/30 sec 257 req/sec

Kernel Cache Enabled 4 clients/160 threads/30 sec 553 req/sec

همانطور که می‌بینید نتیجه فعال سازی کش کرنل پاسخ به بیش از دو برابر درخواست در حالت غیرفعال آن است که یک عدد فوق العاده به حساب میاد.

برای اینکه خودتان هم تست کرده باشید در این آدرس برنامه را دانلود کنید و به دنبال فایل request.cfg بگردید و از صحت پارامترهای server و url اطمینان پیدا کنید. در گام بعدی 5 پنجره خط فرمان باز کرده و در یکی از آن‌ها دستور netsh http show cachestate را بنویسید تا تمامی وروردی‌های entry که در کش کرنل ذخیره شده اند لیست شوند. البته در اولین تست کش را غیرفعال کنید و به این ترتیب نباید چیزی نمایش داده شود. در همان پنجره فرمان wcctl -a localhost -c config.cfg -s request.cfg را زده تا کنترلر برنامه در وضعیت listening قرار بگیرد. در 4 پنجره دیگر فرمان wcclient localhost از شاخه کلاینت را نوشته تا تست آغاز شود. بعد از انجام تست به شاخه نصب کنترلر WCAT رفته و فایل log را بخوانید و اگر دوباره دستور نمایش کش کرنل را بزنید باید خالی باشد. حالا کش را فعال کنید و دوباره عملیات تست را از سر بگیرید و اگر دستور netsh را ارسال کنید باید کش کرنل دارای ورودی باشد.

برای تغییرات در سطح http.sys می‌توانید از ریجستری کمک بگیرید. در اینجا تعداد زیادی از تنظیمات ذخیره شده در ریجستری برای http.sys لیست شده است.