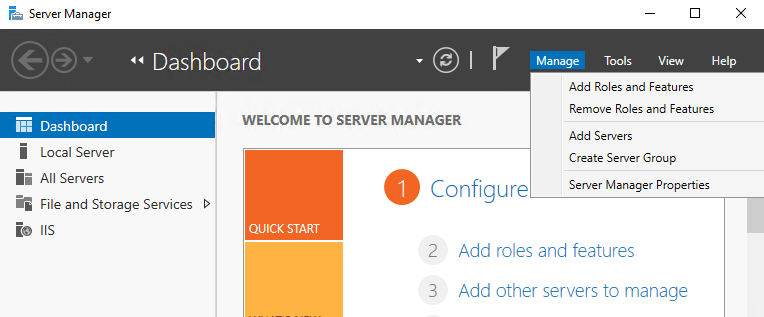
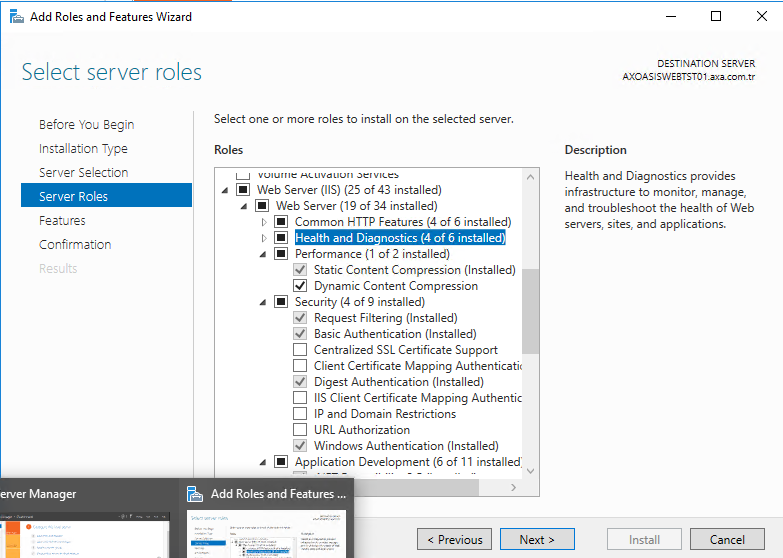
**Kurulum**

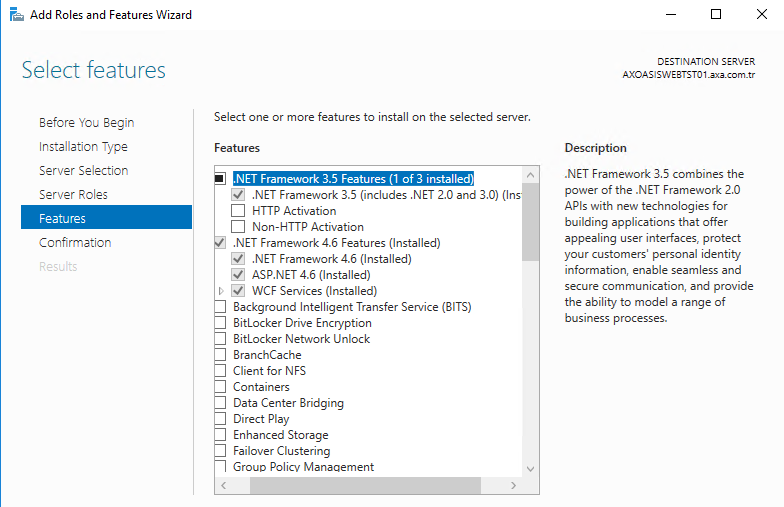
Kurs, Windows Server 2016 makinesi üzerinde gerçekleştirildi. Eklenmesi gereken kurallar ve diğer işlemler Server Manager altında Manage Sekmesinden Add Roles and Features seçilerek seçenekler alanına ulaşılır.



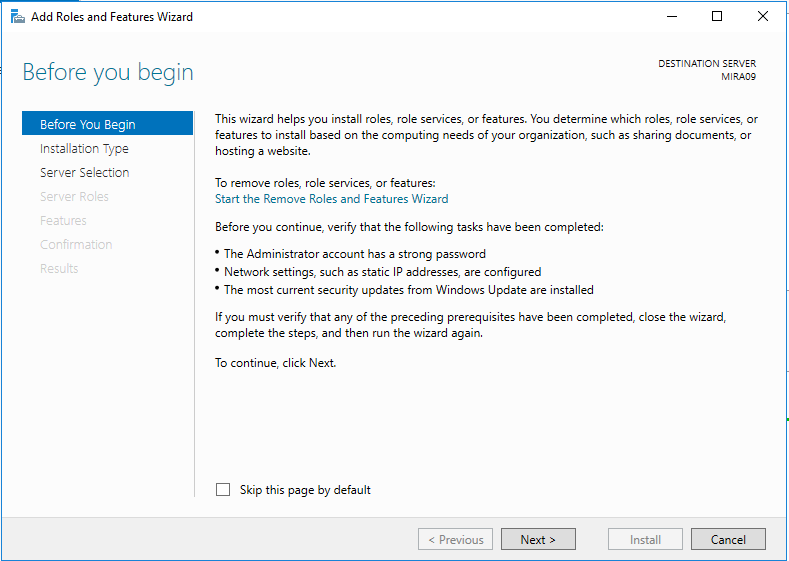
Sonraki aşamalarda Next ile geçilir ve Server Roles sekmesinde eklenebilinecek rol ve özellikler gelir. Buradan Web Server (IIS) sekmesi altındaki özellikler seçilir, Web Server sekmesi altındaki performance ve Application Development altındaki IIS özellikleri seçilir



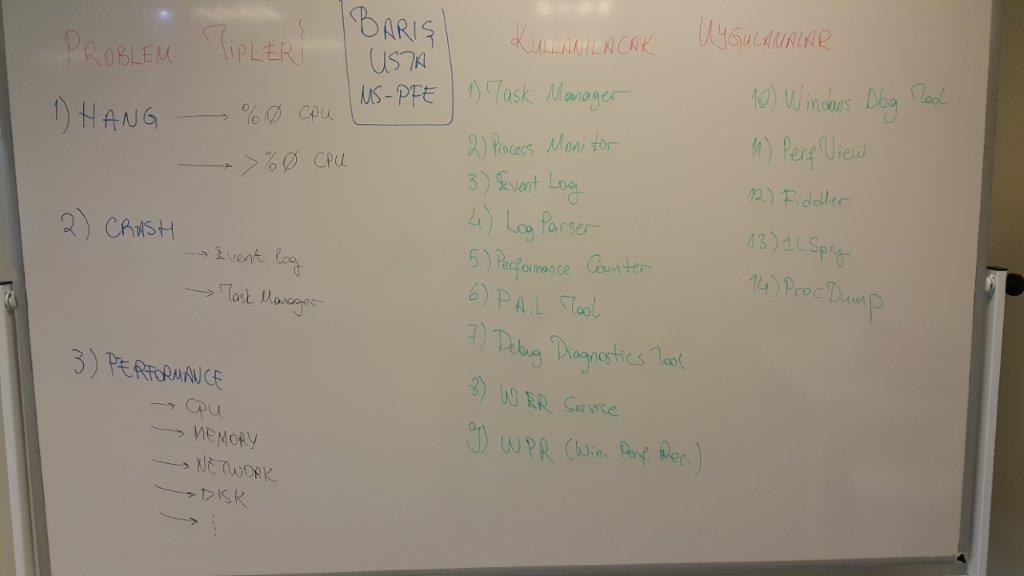
Sonrasında Next diyerek bir sonraki aşamadaki .Net Framework 3.5 & .Net Framework 4.6 özellikleri seçilir.

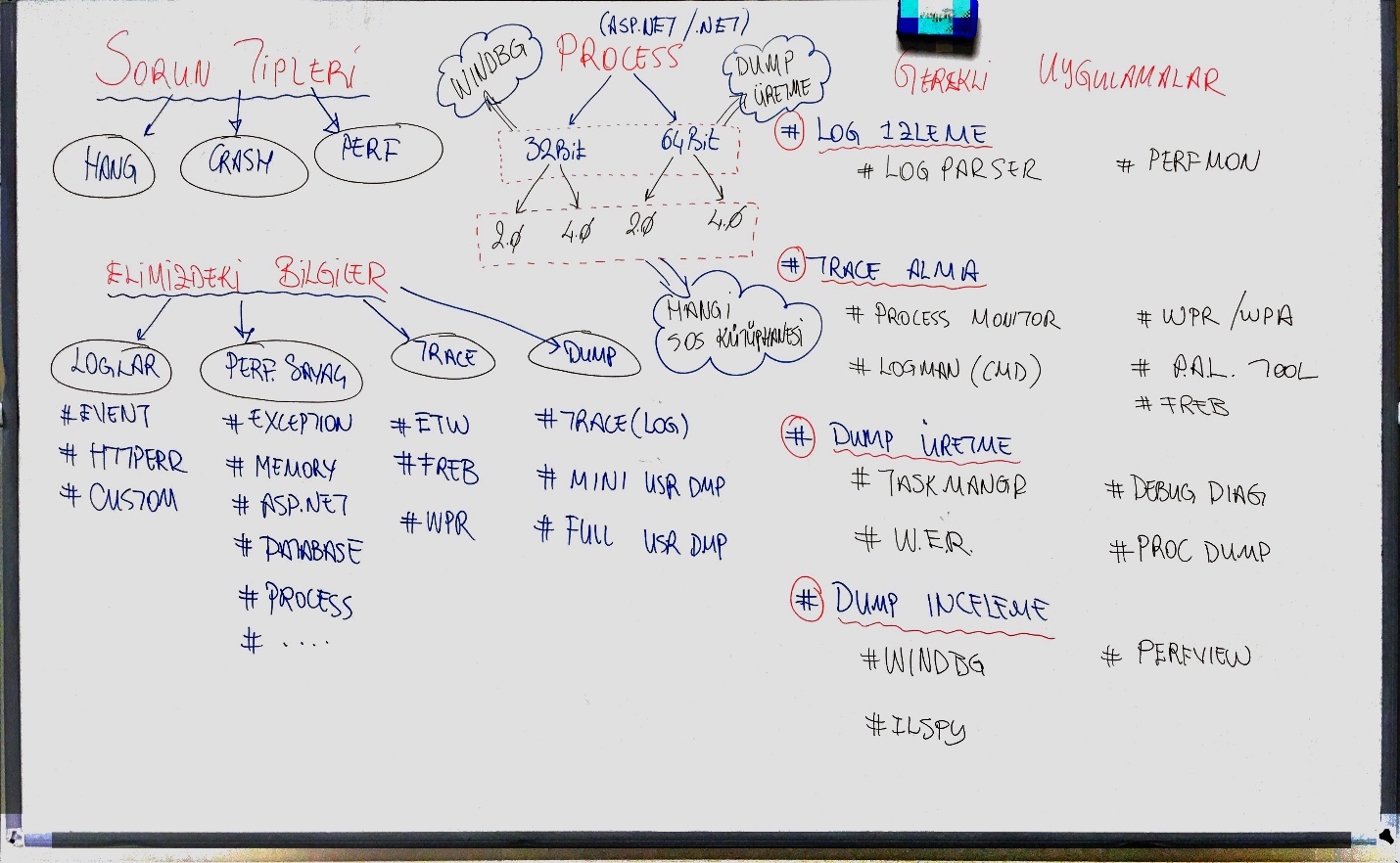


Bir sonraki aşamada, yüklenecek özelliklerin bulunduğu özellikler gösterilir. Install diyerek system üzerine seçili tüm özellikler yüklenir.



**Eğitimin Genel İçeriği**





**BuggyBits Uygulamasının Kurulumu**

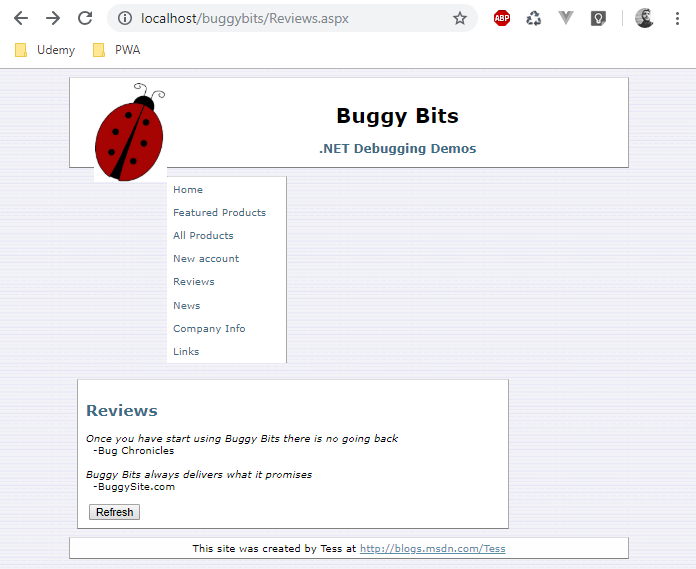
BuggyBits lab boyunca yapılan karşılaşabileceğimiz sorunları test edebilmemizi sağlayan uygulamanın adı. Öncelikle bu uygulamayı sisteme aktarımı için Application Pool oluşturalım ve BuggyBits uygulamasını deploy edelim.

(BuggyBits : <https://blogs.msdn.microsoft.com/tess/2008/02/04/net-debugging-demos-information-and-setup-instructions/>)

**NOT**

1. Bir Application Pool üzerinde 1 tane uygulama koşması tavsiye edilir. Bu sayede hata veya performans konusunda sıkıntı yaşanması durumunda hangi uygulamadan kaynaklandığının tespiti daha kolay sağlanmaktadır.
2. Application Pool kurarken CLR versiyonu .Net 4.0, Pipeline Modu ise integrated seçilmesi gerekir.
3. Application Pool üzerinde, Recycling denilen, poolun kendini kapatıp açtığı bir zaman belirtilmesi gerekir. Gün sonunda veya sistemdeki yoğunluğun en az olduğu zamanda bunun yapılması gerekmektedir.
4. Uygulama publish edilirken web.config dosyası içerisindeki customErrorMode özelliği Readonly yapılması gerekir. Bu sayede client tarafına herhangi bir hata fırlatılmazken, diğer yandan server tarafında hatanın ne olduğu gözlemlenebilir.

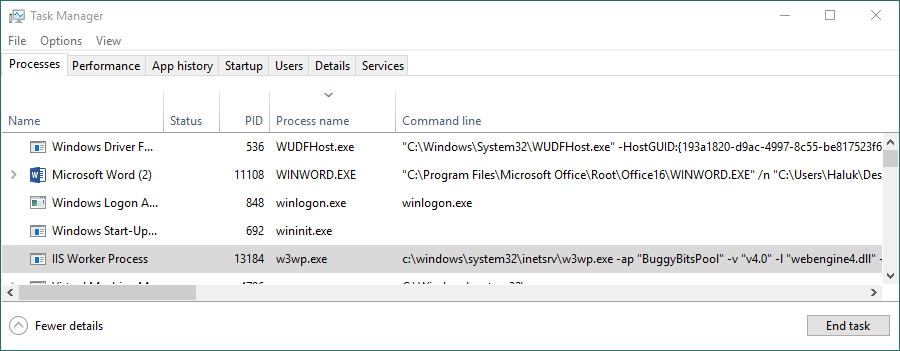
Uygulama kurulduktan sonra BuggyBits uygulaması açılır ve Review sekmesinde refresh butonuna birkaç defa basılır. Gerçekleşen hata senaryosunda uygulamanın crash ettiğini gözlemleyeceğiz. Bu gözlemi yapmadan önce, öncelikle task manager nedir ve niçin kullanılır biraz ondan bahsedelim.



**Task Manager**

Task Manager, çalışan (veya çalışmayan) tüm uygulamalarımızı görebildiğimiz, hangi servislerin çalıştığı, hangi servisin ne kadar kaynak tükettiği ve sisteme maliyetini gözlemleyebildiğimiz uygulamadır.

Task Manager üzerinde çalışan uygulamamızın hangi proccese sahip olup, nerede çalıştığına dair bir çok bilgi vermekte. Bu sayede uygulamada gerçekleşen işlemin hangi thread üzerinde uygulamayı takip edebilir. Herhangi crash olması durumunda uygulamanın process id si ile crashe sebep olan şeyin ne olduğu tespit edilebilir.

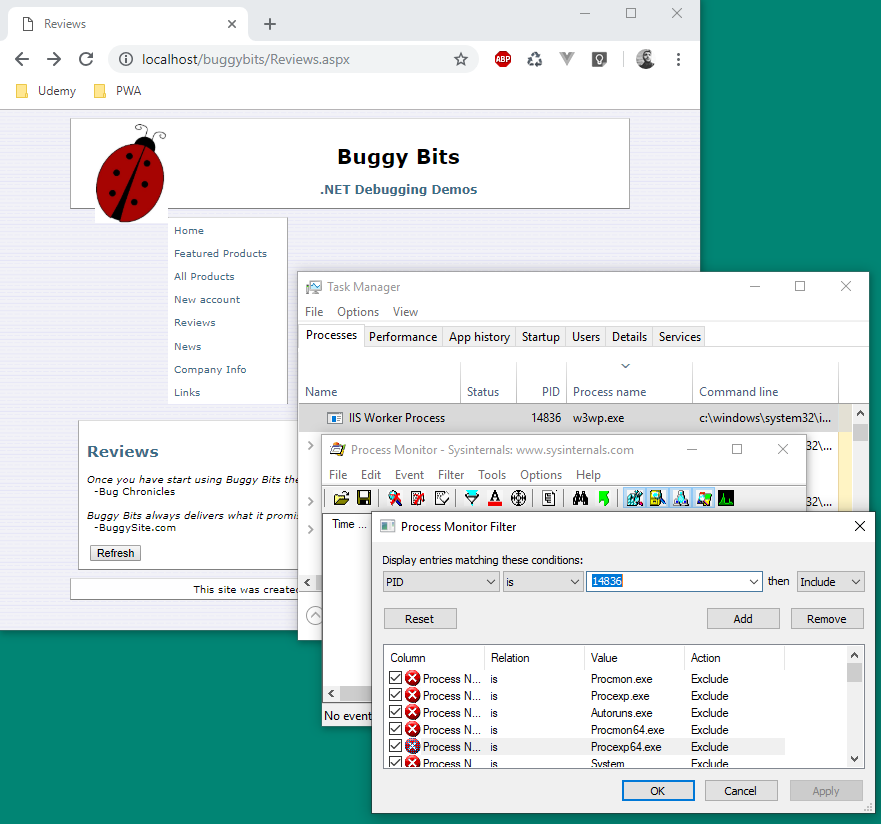


**Process Explorer**

Task Manager’ın alternatifi olan ve task managerdan çok daha güzel özelliklere sahip olan bir tool. Temel olarak hangi processin ne kadar iş yaptığını ve uygulamanın sistemdeki maaliyetini detaylı bir şekilde gösteren programdır.

**Process Monitor**

Process Monitor, tıpkı task manager gibi, işletim sistemi seviyesinde çalışan tüm processleri ve bu processlerin gerçekleştirdiği eventleri bize listeler, aynı zamanda hata gerçekleşmesi durumunda hatanın neden gerçekleştiği ile alakalı bize detay verir. Sysinternals adında bir yazılım şirketi tarafında geliştirilmiştir. Process Monitor uygulamasını



Uygulama tarafına geri dönersek, IIS Worker Process’in üzerinde çalıştığı w3wp.exe sinin process id si 14836. Şimdi Process Monitor uygulamasını açıp, filtre butonuna bastıktan sonra açılan ekranda PID seçilip process id yazılır ve tamam butonuna basılır.

Sonrasında mercek işaretine basılarak process monitor üzerinde ilgili process id ye ait tüm threadler gözlemlenir.

Sonrasında ekranda görünen Refresh butonuna birkaç defa basılır ve HTTP 503 (Service Unavailable) hatası alınır ve sonrasında process monitor ekranında ilgili hatalar incelenir.

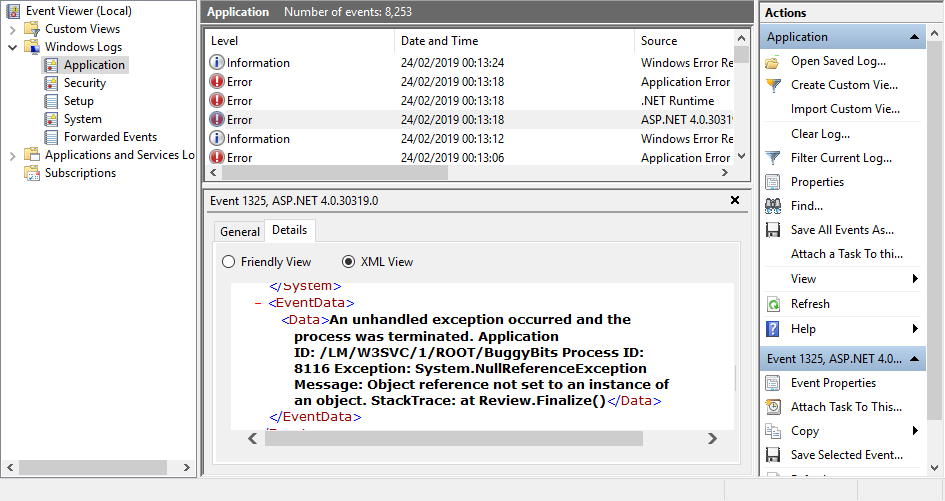
ProcessMonitor kullanılırken dikkat edilmesi gereken en önemli şeylerden biri, process monitor yapılan tüm işlemlerin loglarını tuttuğu için systemi yavaşlatmaktadır. Dolayısıyla kullanılırken dikkat edilmesi gereken en önemli şey, sürekli açık bırakılmaması ve nokta atışı olarak trace edilecek uygulamanın belirlenmesidir.

Diğer yandan hatayı yakalamanın bir diğer yöntemi ise Event Viewer dır.

**Event Viewer**

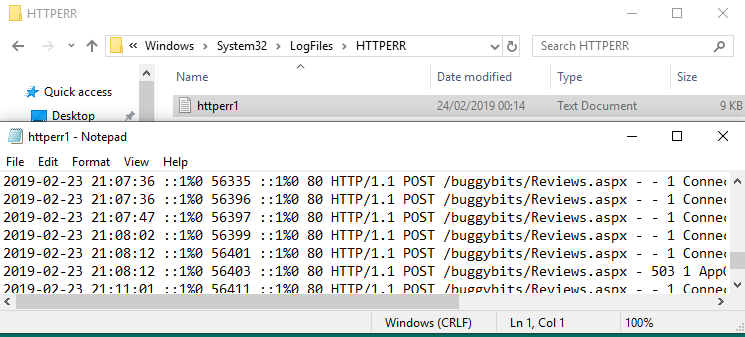
Windowsda hazır olarak gelen bu program ile sistemde gerçekleşen hataları yakalabilmekteyiz. Başlat menüsünden ve çalıştır komutundan eventvwr yazdığımızda gelen bu program sayesinde sistemde gerçekleşen hatayı windows logs sekmesi altından application seçilerek çalışan tüm programların bilgileri gelir. Buradan hatanın detayı, nereden kaynaklandığı bilgilerini elde edebiliriz.

Örneğin, BuggyBits uygulamasındaki hatayı aşağıdaki şekilde gözlemleyebiliriz.



IIS katmanında gerçekleşen hata gerçekleşip gerçekleşmediğinin kontrolünü sadece event viewer programlar ile değil, IIS’in error logları üzerinden de control edebiliriz. Bunun için bakmamız gereken yer

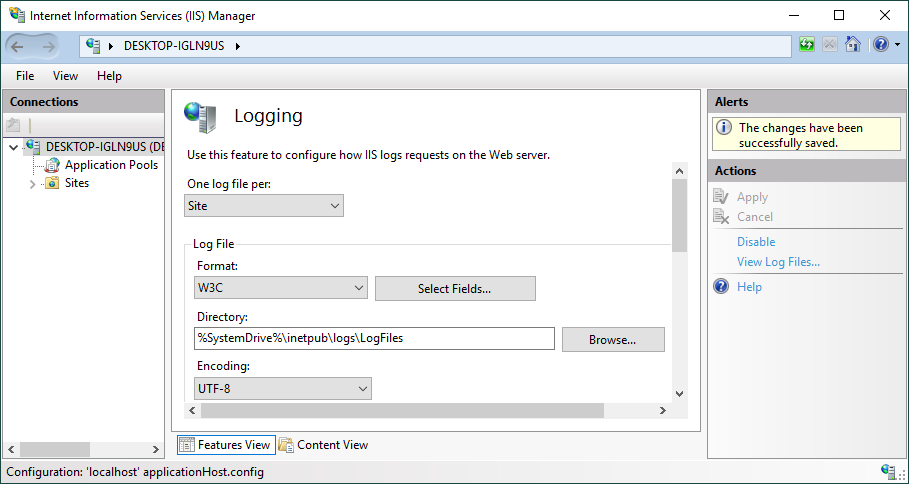
C:\Windows\System32\LogFiles\HTTPERR dizini olacaktır. Buradaki log dosyası zamanla şiştiği için düzenli olarak içerisindeki dosyaları temizlemek ve alınan hataları incelemek gerekir.



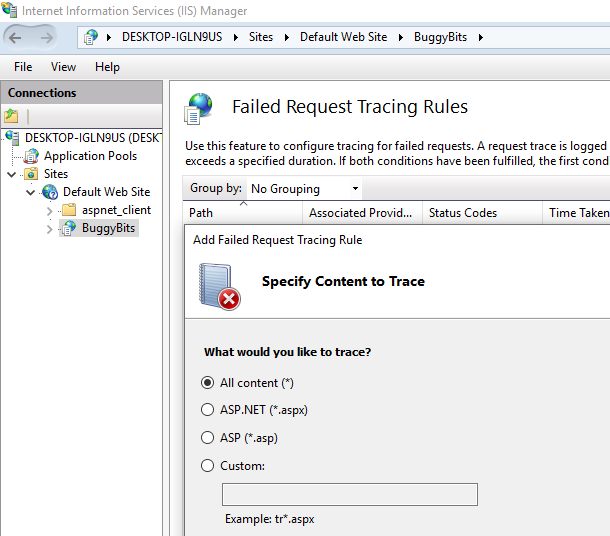
**NOT**

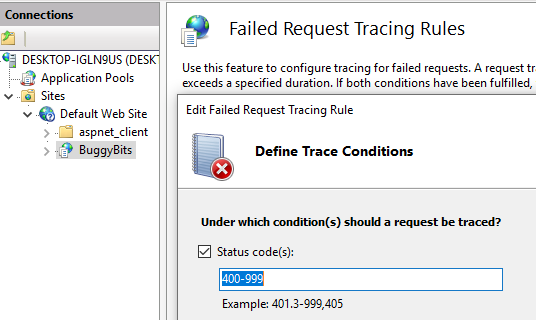
1. IIS üzerinde gerçekleştirilen logların tutulduğu kısım C:\inetpub\logs\LogFiles

Bu kısmı browse diyerek değiştirebiliyoruz.

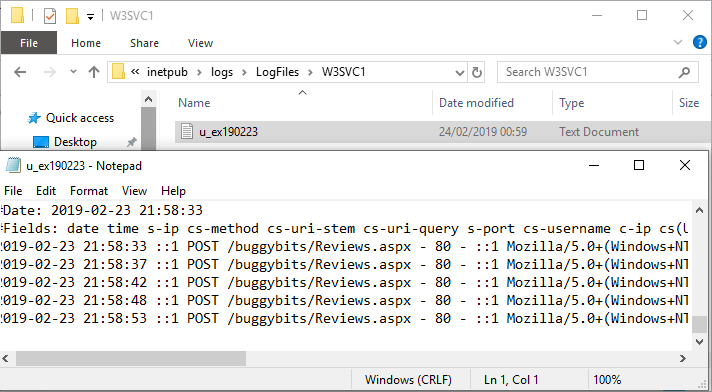


1. Diğer yandan uygulama üzerindeki hataları trace edebileceğimiz gerek sayfa gerek hata kodu bazında yakabileceğimiz durumlarda da yine uygulama üzerindeki Failed Request Tracing Rules özelliğini configure etmemiz gerekiyor. Bunun için Failed Request Tracing Rules sekmesine tıkladıktan sonra yazılacak sayfa(lar) ve (veya) hata türüne (veya türlerine) göre kural atanır.





BuggyBits uygulamasını çalıştırıp Review sayfasından refreshe tekrardan tıklayıp 503 hatası aldığımızda log dosyası C:\inetpub\logs\LogFiles\W3SVC1 altında oluşmaktadır. W3SVC çalışan exenin adı 1 ise IIS üzerinde çalışan uygulamanın ID sidir. Diğer bir önemli konu ise bu log dosyası UTC 0 ile saati loglamaktadır.



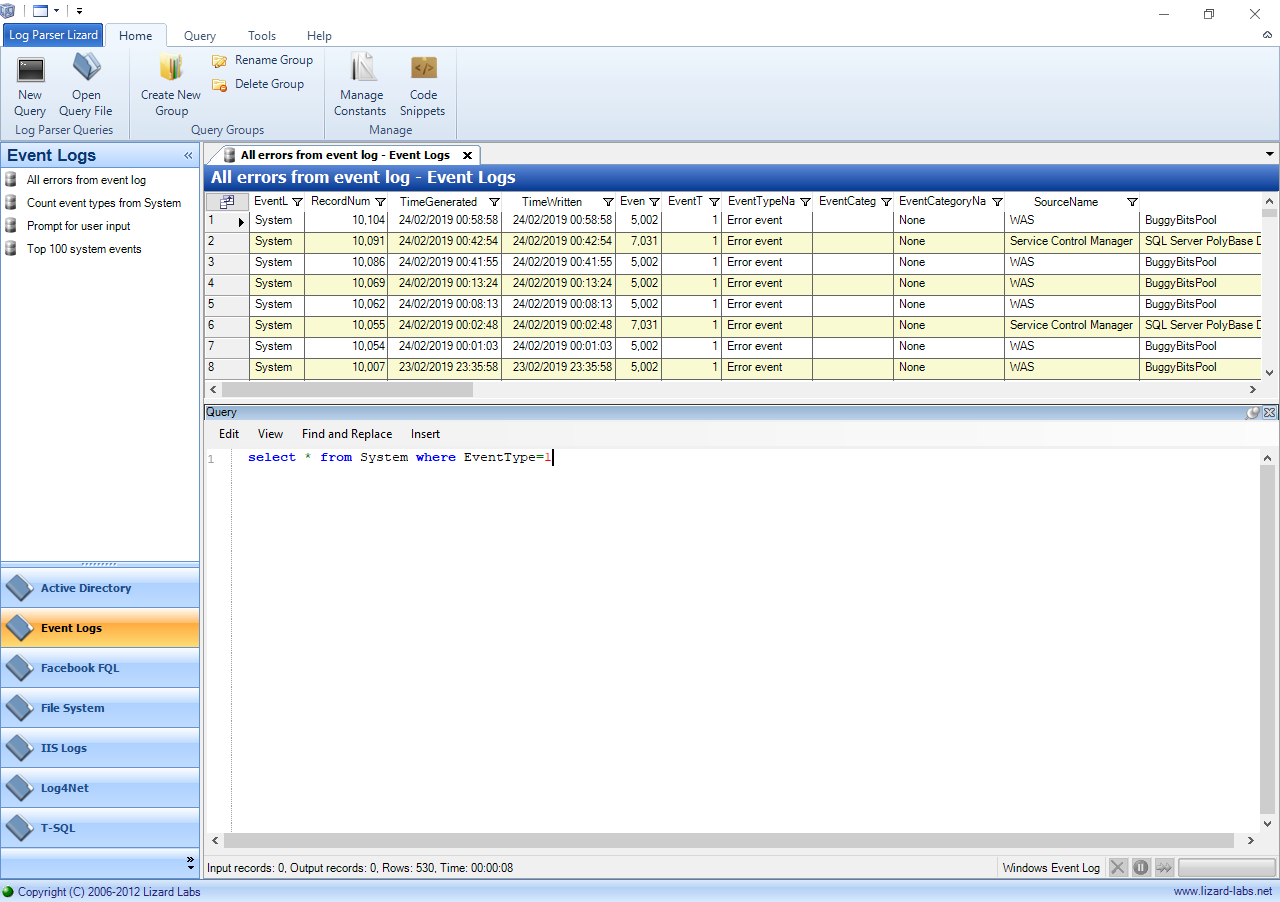
**Log Parser**

Sistem üzerinde tuttuğumuz logları istediğimiz formatta istediğimiz şekilde almamızı sağlayan üzerinde sql sorgusu çalıştırabildiğimiz microsoftun geliştirdiği bir yazılımdır. Sistem üzerinde tutulan logların analizi için log parser kullanılır.

**Log Parser Lizard**

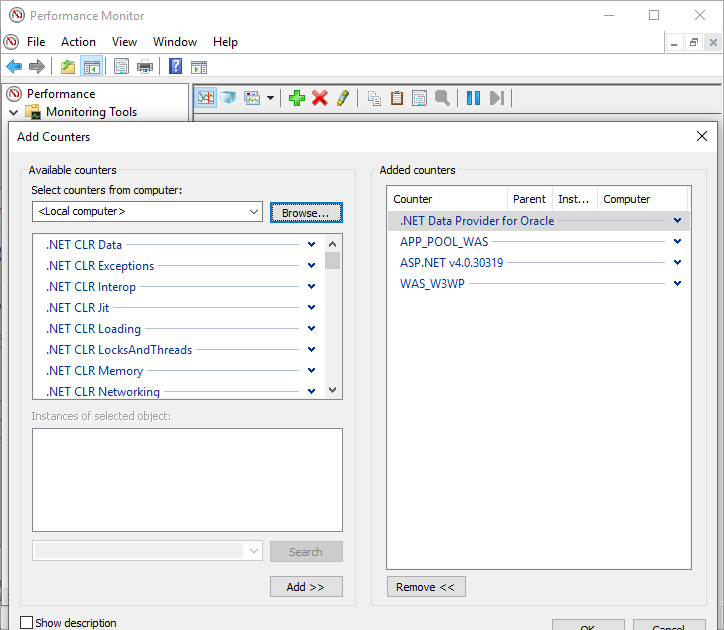
Log Parser uygulamasına arayüz sağlayan kullanışlı bir programdır. Bu program sayesinde sistemde gerçekleşen tüm processlerin durumunu istediğimiz dosya formatında ve istediğimiz şekilde alabiliriz. Bunun için öncelikle yapmamız gereken şey, Log Parser kurduktan sonra bu program kurup, C:\Program Files (x86)\LizardLabs\Log Parser Lizard dizini altında uygulamanın exesini çalıştırmak olacaktır.

Örneğin sistemde alınan tüm hataları aşağıdaki şekilde elde edebiliriz.



**Performance Monitor**

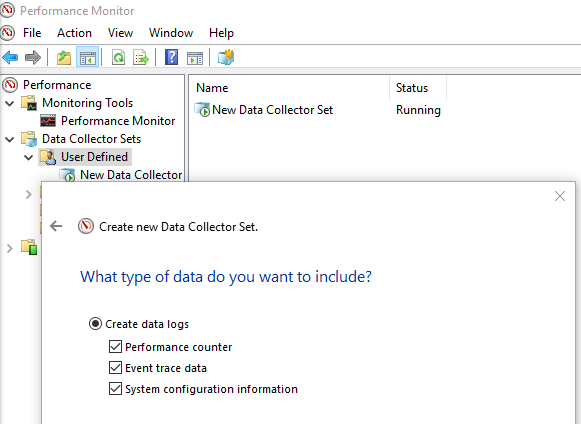
Performance Monitor sistemi takip edeceğimiz, grafiksel olarak işlemcilerin performansı, bellek kullanımı WAS kaynak tüketimi vs. gibi işlemleri gözlemleyebileceğimiz hazır olarak gelen tooldur. Performance Monitorü açmak için başlat menüsünde performance monitor yazdığımızda uygulama ikonu ekrana gelmektedir. Genel görünümü şu şekildedir.



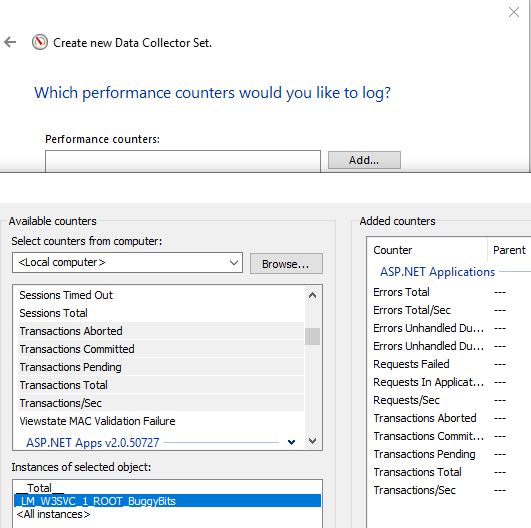
Performance counter ekleyebilmek için, yukarıdaki resimdeki yeşil artı butonuna tıklayarak izleme yapmak istediğimiz özellikleri belirliyoruz.

Performance Monitorun bir diğer güzel özelliği ise, Data Collector Set ile, system üzerinde sadece bizi ilgilendiren kısımları inceleyip belirleyeceğimiz filtrelere göre performans, trace incelemesi yapabiliriz. Bunun için yapmamız gereken şey, Perfomance Monitor Ekranında

Data Collection Sets -> User Defined diyip New Data Collector Set’I seçip Create Manually Dedikten sonra izlemek istediğimiz özellikleri belirtiyoruz.

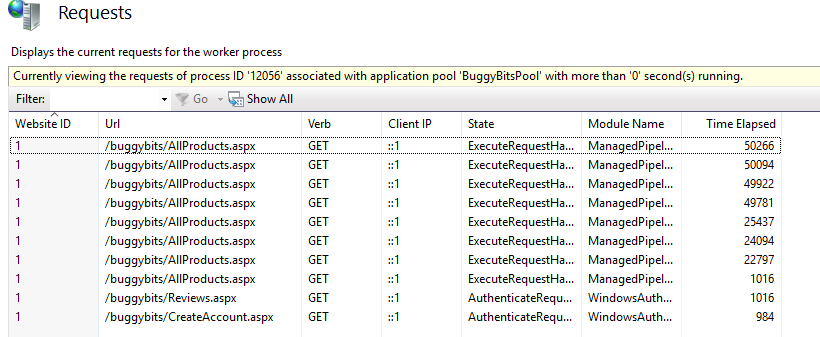


Sonrasında izlemek istediğim alanları belirledikten sonra next diyerek performance counter lar belirlenir. Bu kısımda özellikle takip edilmesi istenen tüm özellikler seçilir ve takip edilir.

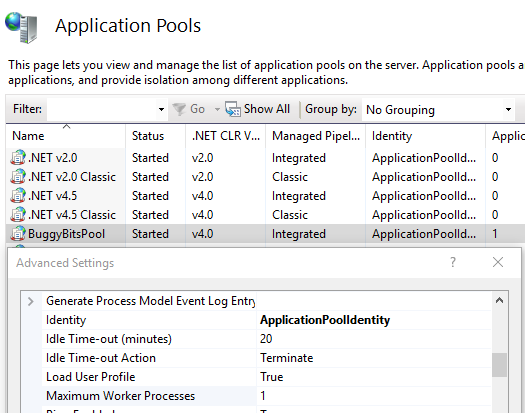


**NOT**

1. IIS üzerinde çalışan uygulama üzerinde yığılma var veya yavaş çalışıyor denildiği anda bakmamız gereken ilk şey, IIS Worker Process olmalı.Worker process de aşağıdaki gibi bir yığılma oluyorsa bu durumda öncelikle yapmamız gereken şey, sistemde bu yavaşlığa sebep olan durumu incelemek



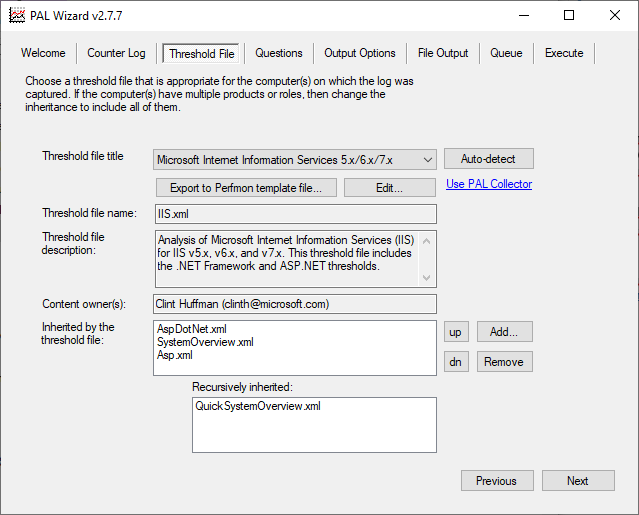
Diğer yandan biraz olsun sistemdeki yığılmayı azaltmak ve yükü hafifletmek için uygulamanın üzerinde bulunduğu Application Pool üzerindeki Maximum Worker Process’i arttırarak biraz olsun sistemdeki yığılmanın önüne geçebiliriz. Fakat bu geçici bir yöntem. IIS gerektiği zaman Worker Process Resource’unu otomatik olarak kendisi ayarlıyor. Fakat bazı durumlarda buna manuel müdahale edilmesi gerekebiliyor.



**Performance Analysis of Logs (PAL)**

PAL, sistemde oluşan bir yığılmanın neden kaynaklandığını tespit edebilmemiz için geliştirilen tooldur. Yaptığı şey, seçilen template’e göre (bu template IIS, Asp.Net, SQL Server, Active Directory vs. olabilir) uygulama üzerinde geliştirme systemin analizini yapıp bize rapor olarak sunan tool.

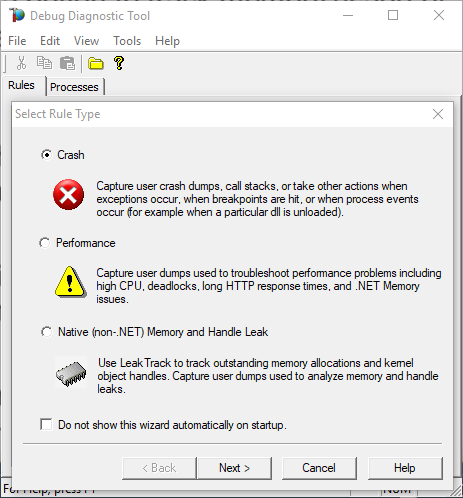
Uygulamanın yüklenmesi çok basit. Next > Next ile geçildikten sonra açılan sayfada Threshold File sekmesine gidilir. Buradan Analizini yapmak istediğimiz uygulama seçilir.



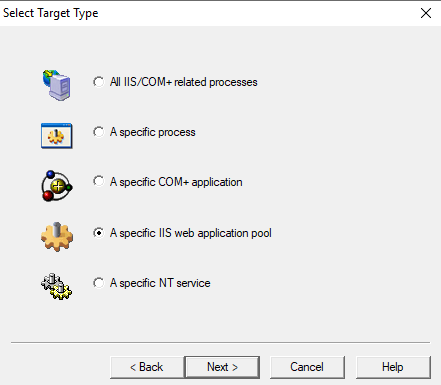
Sonrasında file output simgesinden uygulamanın log dosyasının tutulacağı yer belirlenir ve çalıştırılır. (Detaylı Bilgi : <https://www.youtube.com/watch?v=8fxEs3pPmNI> )

**DebugDiag**

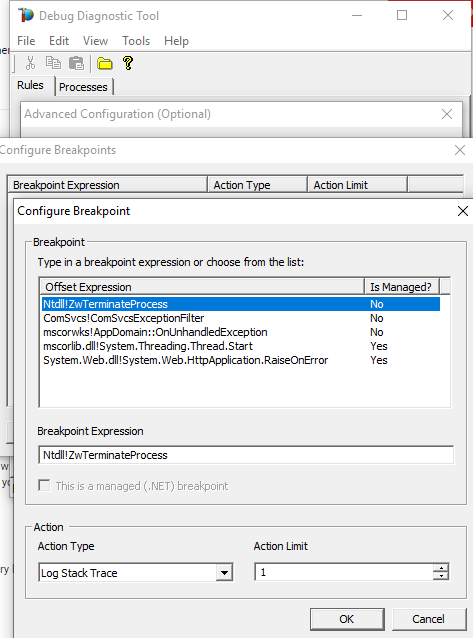
DebugDiag, hata, perforamans veya .Net olmayan objeler üzerinde gerçekleşen hataları yakalayabildiği, bu hataları DebugDiag Analysis ile raporlayıp görüntüleyebildiği microsoftun tooludur. 34 & 64 bit olarak gelir. Çalışan uygulama(lara) göre yüklenmesine dikkat edilmelidir.



Açılan ekranda amacımıza uygun kural seçtikten sonra yapmak istediğimiz işleme uygun process tipi seçilir. Burada uygulamamız IIS Server üzerinde koşuyor.



Sonrasında uygulamanın koştuğu Application Pool seçilir. Ardından Breakpoints kısmına tıklanır ve breakpoints sekmesinden terminate process ile başlayan kod seçilir. Sonrasında full dump seçeneği seçilir ve kural oluşturulur.



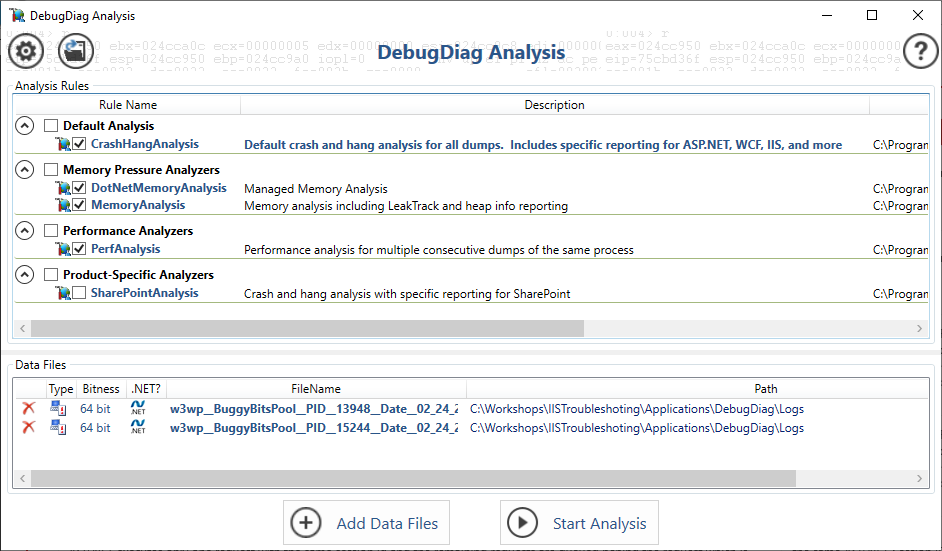
Tanımlanan kuralın ardından hemen aktive edilip edilmeyeceği ve üretilen dump dosyasının pathi belirtiliyor.

Bu aşamadan sonra uygulamaya tekrardan geri dönüş yapıp hataları tekrarlamamız gerekiyor. Hem performans hem de crash sorununu gözlemleyebilmek için uygulamadaki All Products sekmesini başka tablarda defalarca açıyoruz. Diğer yandan Review sekmesindeki Refresh butonuna hata verdirene kadar tıklıyoruz.

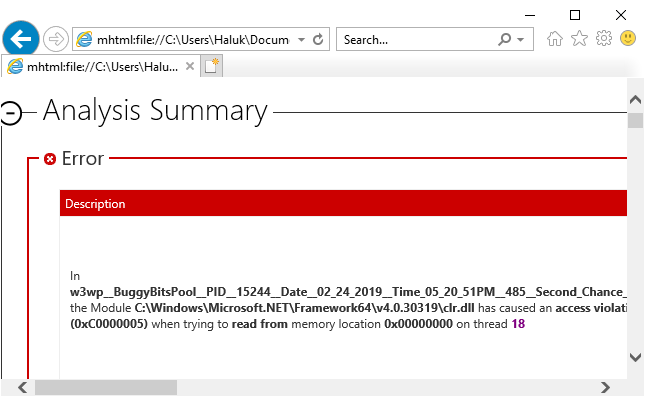
Uygulama üzerinde gerçekleşen hataları DebugDiagAnalysis toolu üzerinden inceleyeceğiz.

**DebugDiag Analysis**

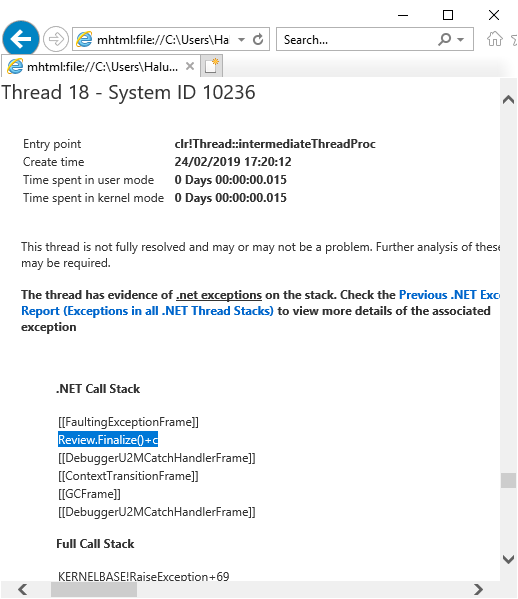
Elde ettiğimiz dump dosyalarını kontrol edebileceğimiz, birden fazla dump dosyasını ekleyip analiz edebileceğimiz tooldur. DebugDiag birden fazla dump dosyasını analiz edebilme imkanı vermekte. DebugDiag’ın ürettiği dosyaları dosyaları seçip analiz işlemine başlayalım.



Dosyaları ekledikten sonra start analysis diyerek dosyaların analizine başlanır. Analiz sonucunda elde edilen rapor aşağıdaki gibidir.



Hatanın thread 18 de verdiğini belirtmekte. 18 rakamının üzerine tıkladığımızda hatanın hangi metodda gerçekleştiğini gözlemlemekteyiz.

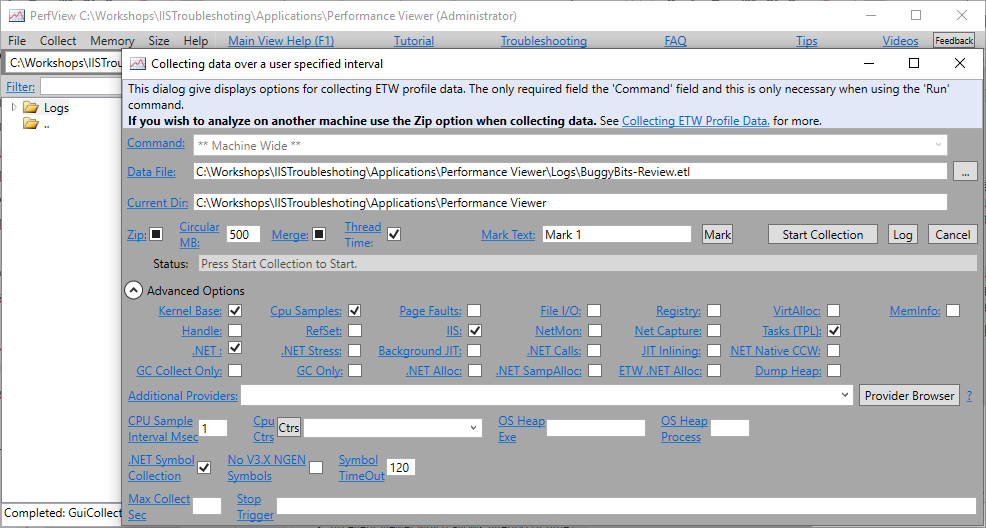


Hata & performans tespiti için kullandığımız bu toolların yanı sıra sistemde (veya uygulamada) gerçekleşen hatayı yakalayabilmek, tabiri caize nokta atışı hatanın nereden kaynaklandığını gözlemleyebilmek için kullandığımız birkaç tool var. Şimdi de bunları tanıyalım.

**PerfViewer**

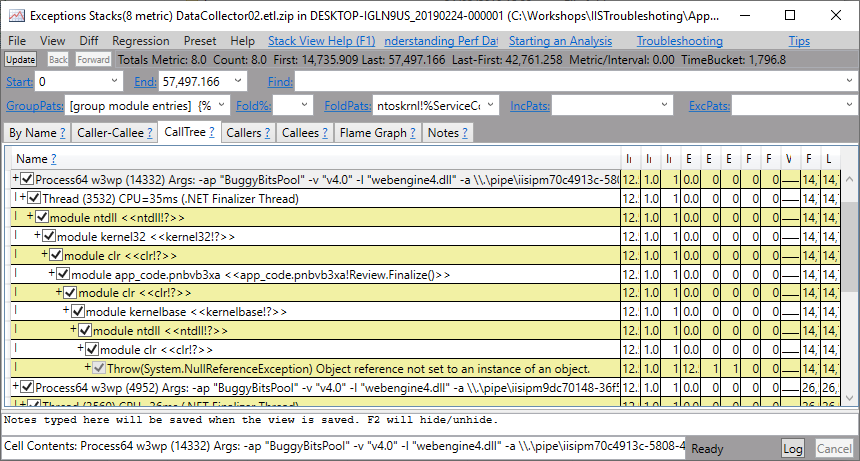
Bu uygulama ile BuggyBits uygulamasında gerçekleşen hatayı yakalayalım. Review sayfasına gidip refresh butonunda aldığımız HTTP 503 hatasının neyden kaynaklandığını tespit edelim. Bunn için öncelikle PerfViewer üzerinde bazı düzeltmeler yapmamız gerekmekte.

Açılan uygulamada üst sekmede yer alan Collect butonuna bastıktan sonra açılan alt menüden tekrardan collect seçilir. Burada açılan ekran üzerinde thread time ve IIS seçenekleri seçildikten sonra oluşturulan log dosyasının nereye yazılacağı belirlenir.



Sonrasında Start Collection diyip loglama işlemini başlatıyoruz. BuggyBits sayfasındaki Refresh butonuna hata alıncaya kadar basıp sonrasında loglama işlemini durduruyoruz. Ardından log dosyası üzerinden yapılan eventleri gözlemliyoruz.

Bildiğimiz üzere, IIS üzerine kurulan bir uygulama w3wp.exe yi kullanmakta. Dolayısıyla w3wp.exe üzerindeki processler üzerinde hata alan işlemleri gözlemlediğimizde



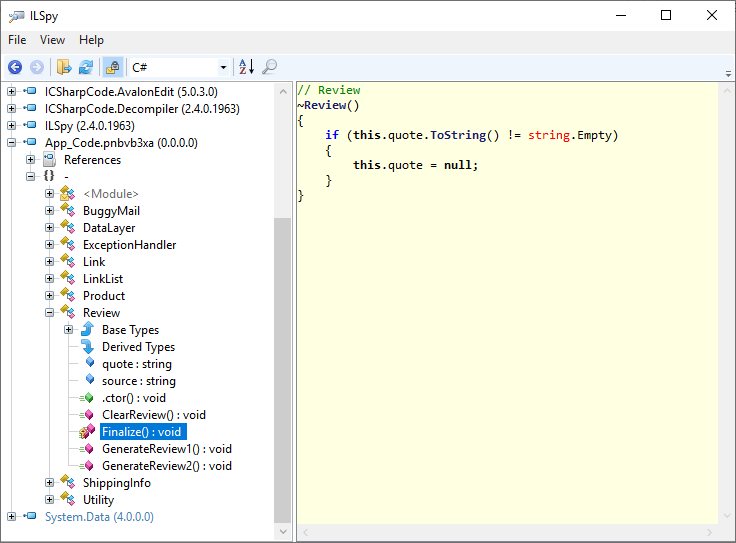
Buradan gözlemlediğimiz şey, Review.Finalize() (yani deconstruction methodunda) “Object ref. not set to instance of an object” hatası almaktadır.

Şimdi de yukarıda yakaladığımız bu hatanın nereden kaynaklandığını yakalayabilmek için ILSpy programını kullanacağız.

**ILSpy**

ILSpy, compile edilmiş dll’i incelememizi sağlayan bir tool. Basit ve kullanışlı. Herhangi bir kuruluma ihtiyaç duymamakla birlikte exe yi çalıştırıp ilgili dll’i sürükle-bırak ile inceleyeceğimiz tooldur.

Yukarıdaki örneğe tekrar geri dönelim. Hatayı tespit ettik, şimdi de Review classı içerisindeki deconstruction metodunu incelemeye geldi. Bunun için IIS üzerinde çalışan dll’i almamız gerekiyor. Bunun için ise C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Temporary ASP.NET Files\ dizini altında hangi uygulama üzerinde işlem yapılacaksa o seçilir, burada belirtilen dll seçilerek uygulamanın içerisine atılır. Yukarıda perfview için yaptığımız örnekte, App\_Code.pnbvb3xa.dll içerisinde hata aldığını görmekteyiz. İlgili dll’i ILSpy üzerine attığımızda aşağıdaki hata ile karşılaşmaktayız.



Hatanın this.quote objesinin boş gelmesinden kaynaklandığı gözlemlenmektedir.