

Технологии разработки Internet- приложений

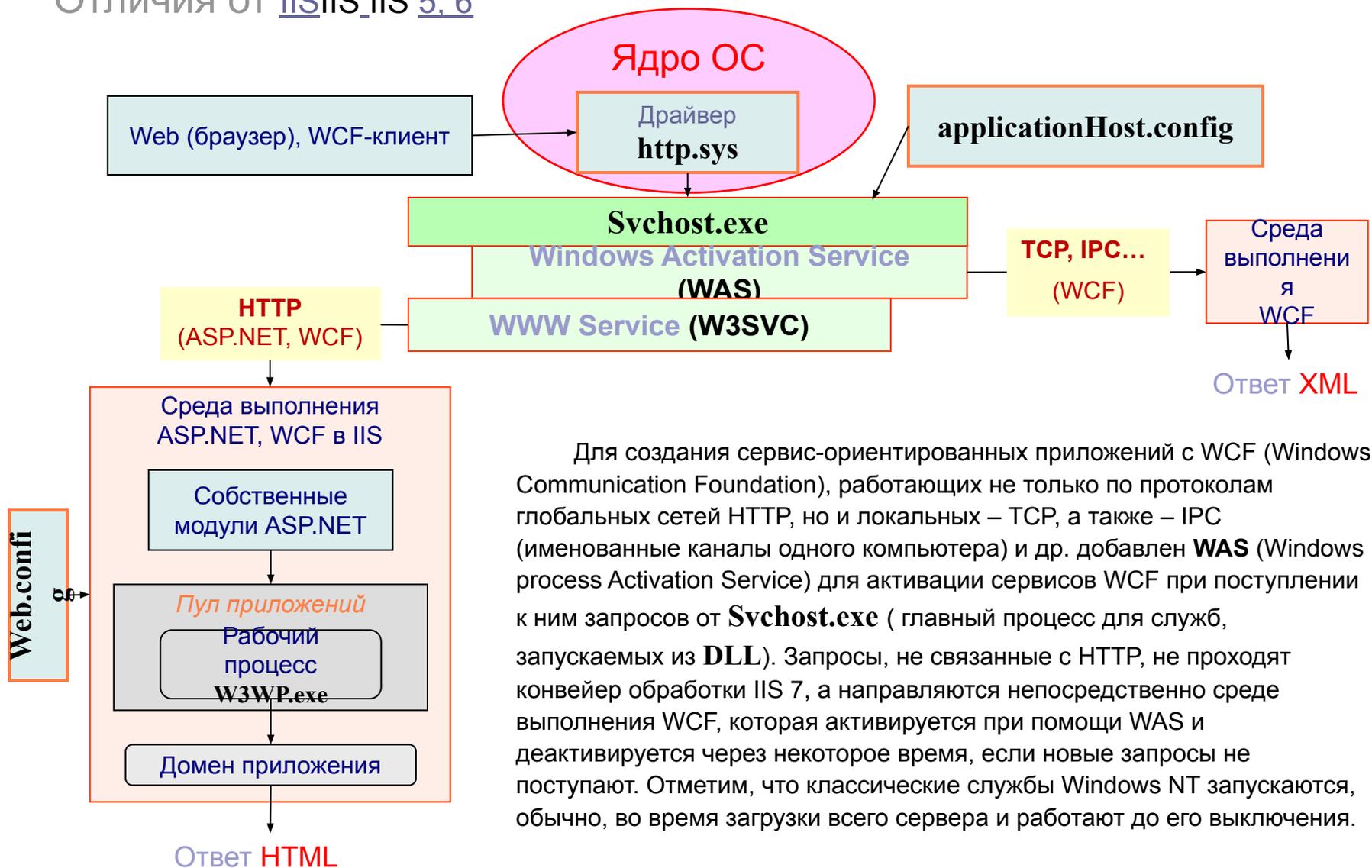
Архитектура и администрирование IIS 7

исполняющая среда ASP.NET в IIS 7,
администрирование, конфигурирование, пулы IIS 7

(архитектура(архитектура, администрирование IIS 5, IIS 6)

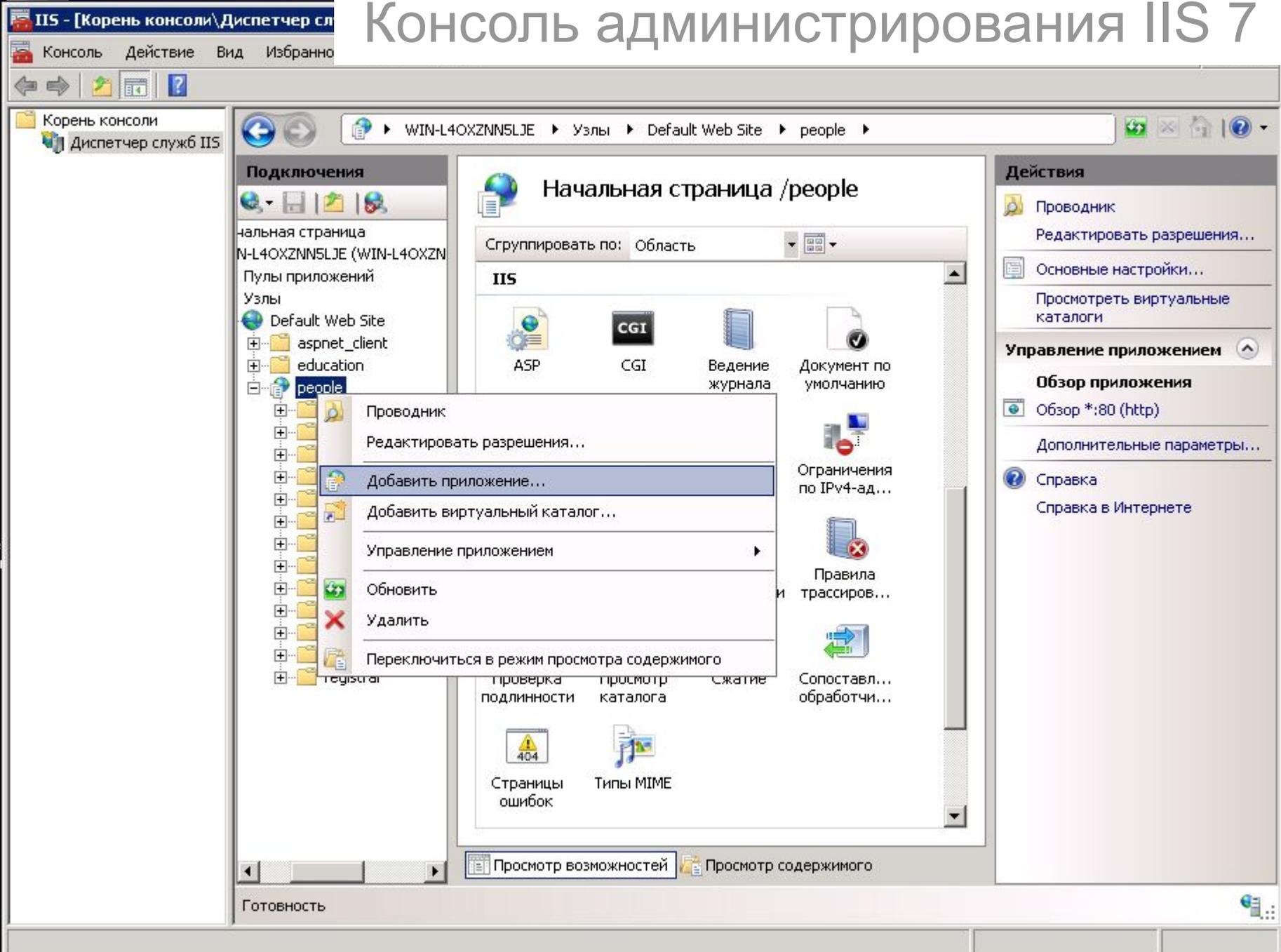
Исполняющая среда ASP.NET в IIS 7

Отличия от IIS 5.6



Для создания сервис-ориентированных приложений с WCF (Windows Communication Foundation), работающих не только по протоколам глобальных сетей HTTP, но и локальных – TCP, а также – IPC (именованные каналы одного компьютера) и др. добавлен **WAS** (Windows process Activation Service) для активации сервисов WCF при поступлении к ним запросов от **Svchost.exe** (главный процесс для служб, запускаемых из **DLL**). Запросы, не связанные с HTTP, не проходят конвейер обработки IIS 7, а направляются непосредственно среде выполнения WCF, которая активируется при помощи WAS и деактивируется через некоторое время, если новые запросы не поступают. Отметим, что классические службы Windows NT запускаются, обычно, во время загрузки всего сервера и работают до его выключения.

Консоль администрирования IIS 7



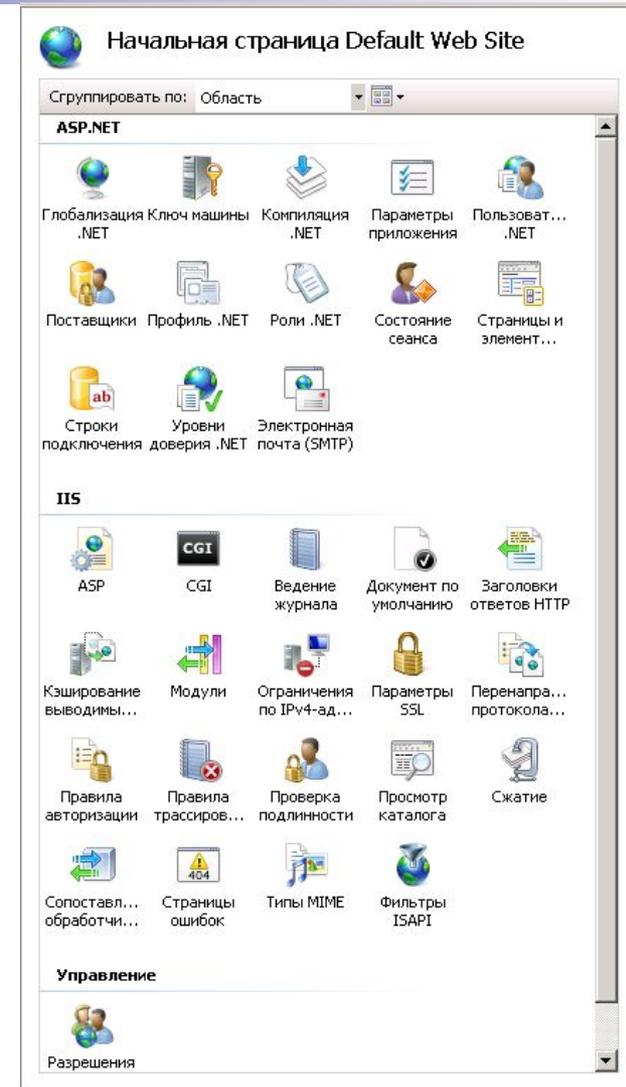
Особенности IIS 7

IIS 7.0 разбивает веб-сервер (IIS 6) на небольшое ядро сервера (**Svchost.exe**) и более чем 40 собственных модулей-компонентов (см. рис.)

Модули – это ASP.NET компоненты, используемые сервером для обработки запросов вместо **aspnet_isapi.dll**. Например, для проверки подлинности учетных данных клиентов используются модули проверки подлинности, для управления кэшем – модули кэширования и т. д. Необходимые модули встраиваются непосредственно в *HTTP-конвейер обработки запросов*, при этом сервер приобретает строго заданную и не избыточную функциональность. Преимущества модульной конструкции – это уменьшение количества возможных направлений атак и размера веб-сервера.

Хранилище конфигураций IIS 6 – *метабаза* заменена на файл конфигурации `windows\system32\inetsrv\config\applicationHost.config`, в котором содержатся значения по умолчанию глобальных настроек сервера, и на файлы **web.config**, находящиеся в структуре каталогов приложения.

Локальные файлы `web.config` в IIS 7 инкапсулируют (могут делегировать от `applicationHost.config`) требуемую настройку сервера и приложения в структуру каталогов приложения. Это существенно упрощает развертывание, которое сводится просто к копированию приложения в каталог на целевом сервере.



Пример web.config настроек IIS 7

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.web>
    <customErrors mode="Off" />
  </system.web>
  <system.webServer>
    <directoryBrowse enabled="true" />
  </system.webServer>
  <runtime>
    ...
  </runtime >
  ...
</configuration>
```

Параметры приложения

Параметры делегирования

Параметры настройки CLR .NET Framework

Настройки реализуются через модуль IIS –

Configuration Editor

«Глобальные» изменения сервера

ApplicationHost.config

- Параметры уровня сервера
- Делегирование глобального компонента
- Пользователи диспетчера IIS

Отдельные изменения узла

Web.Config Site #1

- Параметры делегирования
- Параметры приложения
- Параметры .NET

Web.Config Site #2

- Параметры делегирования
- Параметры приложения
- Параметры .NET

Web.Config Site #3

- Параметры делегирования
- Параметры приложения
- Параметры .NET

Система конфигурации позволяет настраивать приложения путем **делегирования** разделов из **applicationHost.config** в **Web.config**. По умолчанию разделы конфигурации IIS заблокированы и могут быть настроены только в **applicationHost.config** за исключением разделов:

Документ по умолчанию, Просмотр каталога, Протокол HTTP, Перенаправление HTTP.

При необходимости можно разблокировать дополнительные разделы конфигурации через диспетчер IIS

Объекты конфигурации: веб-сервер, узел (сайт), приложение, виртуальный или физический каталог, файл.

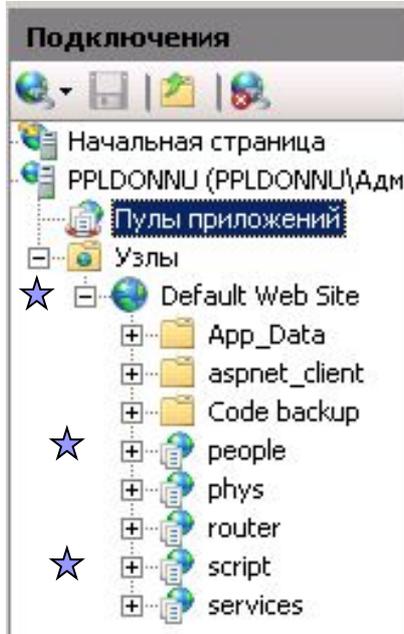
* - в интегрированном режиме IIS 7 (см. далее) этот элемент используется только для управления пулом.

Элементы конвейера HTTP в IIS 7

Компоненты конвейера обработки запросов IIS 7 (см. подробнее [Модель процесса ASP.NET, конвейер HTTP.ppt](#)) включают в себя:

- **пулы** приложений,
- **модули** IIS – компоненты сервера,
- **фильтры** ISAPI,
- **обработчики** для файлов .asp, .aspx...
- **пользовательские обработчики** в файлах .asp, .aspx...

Пулы IIS 7



Пулы приложений

Эта страница позволяет просматривать и изменять список пулов приложений, существующих на сервере. Каждый пул приложений связан с рабочим процессом, содержит одно или несколько приложений и обеспечивает изоляцию приложений друг от друга.

Фильтры:

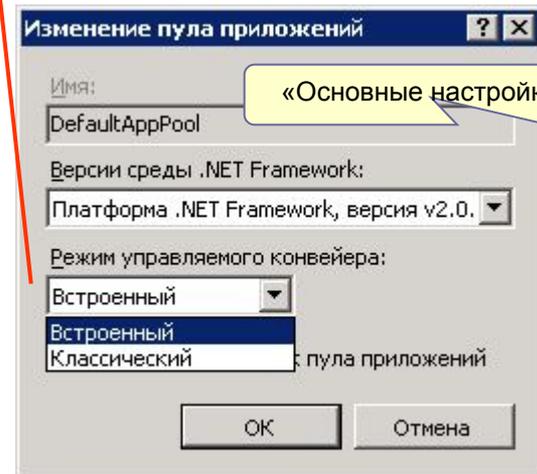
Имя	Состоя...	Ве...	Режим упра...	Удостоверение	Приложения
Classic .NET App...	Работает	v2.0	Классический	NetworkService	0
DefaultAppPool	Работает	v2.0	Встроенный	NetworkService	3
phys	Работает	v2.0	Встроенный	NetworkService	1
router	Работает	v2.0	Встроенный	NetworkService	1
services	Работает	v2.0	Встроенный	NetworkService	1

★ Три приложения работают в пуле по умолчанию: **корень сервера, people, script**.
Остальные – в отдельных именных пулах

Пулы работают от имени **NetworkService**

Пулы приложений в IIS 7 работают в одном из двух режимов: **встроенный (интегрированный)** и **классический (ISAPI)**.

Если приложение запущено в пуле, работающем в интегрированном режиме, то IIS применяет свои модули совместно с приложением ASP.NET, т.е. приложения интегрируются со всем веб-сервером и запросами к нему. Теперь можно управлять всем IIS 7 из приложений ASP.NET. Если приложение запущено в пуле с классическим режимом, то веб-сервер направляет запросы через **aspnet_isapi.dll**, т.е. обрабатывает их аналогично IIS 6.0.



Основные учётные записи для пулов IIS 7

- **IIS_IUSRS** – группа, члены которой могут запускать рабочий процесс ASP.NET в IIS 7 с минимальными правами.
- **IUSR** – учётная запись для анонимного доступа к Web-сайтам в IIS 7. Она входит в **IIS_IUSRS**.
- **NetworkService** – системная учётная запись от имени которой выполняются рабочие процессы ASP.NET в IIS 7 а также все приложения и системные процессы, запущенные рабочим процессом. Имеет ограниченные права на локальном компьютере, а также может использоваться для доступа к ресурсам в сети Active Directory на основании учетной записи компьютера.
- **ApplicationPoolIdentity** – виртуальная учетная запись удостоверения пула приложений в **IIS 7.5**. Для каждого пула получает своё реальное имя, совпадающее с именем пула. Создается автоматически, при создании каждого нового пула, и рабочий процесс пула запускается под этой учетной записью. Запись имеет самые минимальные права на локальном компьютере. Это наиболее безопасный вариант.

Другие записи, которые можно использовать в IIS:

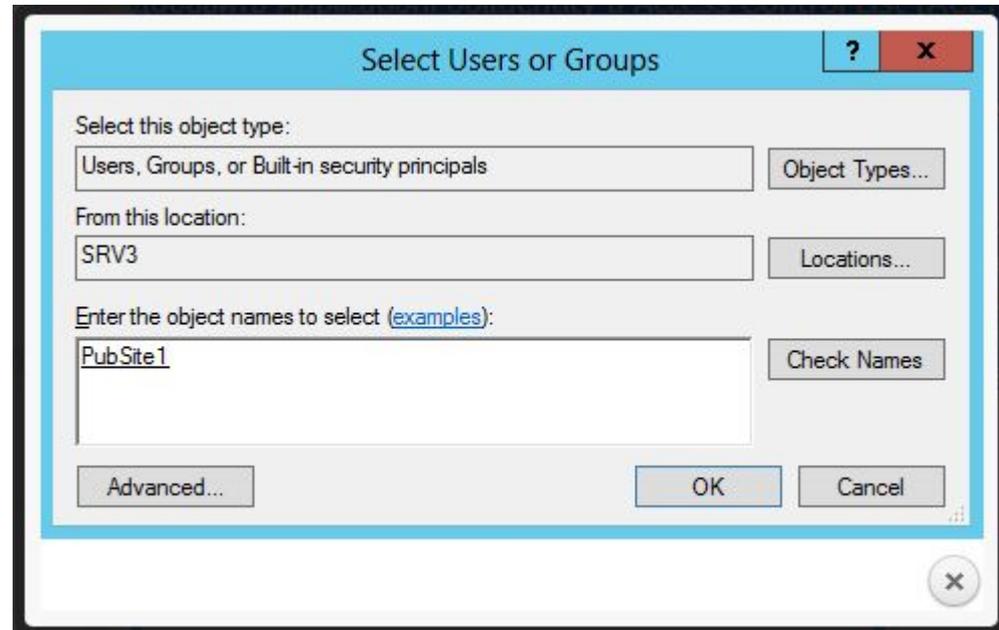
- **LocalService** – системная учётная запись, которая имеет ограниченные права на локальном компьютере. Примерно то же самое, что и NetworkService, но ограничена только локальным компьютером.
- **LocalSystem** – системная учётная запись, имеющая неограниченные права на локальном компьютере. Наименее безопасный вариант, по возможности не рекомендуется использовать эту запись в IIS.

Рабочие процессы IIS можно запускать и от других учётных записей с использованием [олицетворения \(имперсонализации\)](#).

Настройка доступа к ресурсам на основе пула приложений

Иногда Web-приложению может потребоваться доступ к определенной папке или файлу на диске. Чтобы добавить **ApplicationPoolIdentity** в Access Control List (ACL) необходимо:

- Запускаем Windows Explorer;
- Выбираем нужный файл или директорию, кликаем по ней правой клавишей мыши и выбираем пункт Свойства (Properties);
- Переходим на вкладку Безопасность (Security),
- Кликаем по кнопке Изменить (Edit),
затем Добавить (Add);
- В поле Размещение (Locations) выбираем локальную машину;
- Вводим имя пользователя в виде **"IIS AppPool\имя пула приложений"**.
Так для пула приложений PubSite1 имя пользователя будет выглядеть **"IIS AppPool\PubSite1"**;
- Проверяем имя клавишей Проверить имена (Check Names) и жмем OK.



Модули в IIS 7

Модули делятся на три типа: **собственные** (машинные), **управляемые** (управляются средой .NET Framework, см. [Модули HTTP, события приложения - Global.asax.ppt](#)) и **неуправляемые** (приложения Win32 для IIS 5, 6, которые не управляются средой .NET Framework).

Собственные модули – это собственный код библиотек DLL веб-сервера, который выполняет специфическую работу по обслуживанию запросов. Например, WindowsAuthModule выполняет интегрированную проверку подлинности Windows с использованием NTLM.

Модуль HTTP управляемого кода можно задать как файл класса в папке **App_Code** приложения. Также можно создать модуль как проект библиотеки классов, [скомпилировать его и добавить](#) в папку **Bin** приложения. Эти модули могут выполнять работу по обработке запроса параллельно с собственными модулями. Управляемые модули позволяют создавать уникальные серверы и управлять их работой. Например, можно реализовать проверку подлинности с помощью форм не только с содержимым ASP.NET, но и другими типами, например, PHP.

Модули, зарегистрированные в элементе **globalModules** файла **ApplicationHost.config**, обладают глобальной областью действия для всех веб-приложений. Если глобальный модуль в веб-приложениях не используется, его можно отключить через диспетчер IIS.

Фильтры ISAPI в IIS 7

[Фильтры ISAPI](#) – это программы, которые можно добавить в IIS для улучшения работы веб-сервера. В службах IIS 7 модули HTTP и обработчики событий в **Global.asax** заменяют функциональные возможности, которые обеспечивали фильтры ISAPI в службах IIS 6.0. Однако все еще можно добавлять фильтры ISAPI, если необходимо использовать предоставляемые ими функциональные возможности.

Обработчики запросов в IIS 7

В службах IIS 7 обработчики формируют ответы на запросы к узлам и приложениям. Как и модули, обработчики внедряются как **собственный** или как **управляемый** код. При наличии определенного типа содержимого на узле или в приложении необходимо иметь обработчик, который сможет обработать запрос к этому типу содержимого. Следует сопоставить этот обработчик с типом содержимого.

Например, обработчик страниц ASP реализован в модуле **IsapiModule** и по умолчанию включён (колонка «Состояние» на рисунке) в службы IIS 7. Он сопоставляется (колонка «Путь» на рисунке) со всеми запросами к **.asp**-файлам.

Для внедрения **неуправляемых** кодов обработчиков CGI и ISAPI версий IIS 5,6 необходимо (см. следующие слайды) сначала в заданных каталогах приложения включить собственные модули IIS 7: CGI-exe, ISAPI-dll для поддержки неуправляемых кодов. Далее можно добавить и включить выполнение файлов неуправляемых обработчиков.



Сопоставления обработчиков

Эта функция предназначена для указания ресурсов (библиотек DLL и управляемого кода), которые обрабатывают определенные типы запросов.

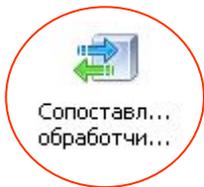
Имя	Путь	Состояние	Ти...	Обработчик
Отключен				
CGI-exe	*.exe	Отключен	Файл	CgiModule
ISAPI-dll	*.dll	Отключен	Файл	IsapiModule
Включен				
ASPClassic	*.asp	Включен	Файл	IsapiModule
AssemblyResourceLoade...	WebReso...	Включен	Не ...	System.Web.Ha
AXD-ISAPI-2.0	*.axd	Включен	Не ...	IsapiModule
HttpRemotingHandlerFac...	*.rem	Включен	Не ...	System.Runtime

Установка модуля FastCGI позволяет подключать CGI-обработчики PHP, Perl, Python... Модуль FastCGI допускает повторное использование процесса благодаря поддержке пула рабочих процессов, таким образом обеспечивая значительно большую производительность. Скопировать и настроить модуль PHP для IIS 7 можно, например, по адресу <http://msdn.microsoft.com/ru-ru/magazine/cc135973.aspx>.

Запуск приложений в IIS 7

Шаг 1 – создать в консоли IIS *приложение* или *виртуальный каталог* для хостинга неуправляемого (не .NET) CGI-ISAPI обработчика запросов.

Шаг 2 – в созданном приложении при помощи «Сопоставления обработчиков» разрешить управляемым модулям CgiModule и IsapiModule исполнять неуправляемые CGI и ISAPI



Сопоставления обработчиков

Эта функция предназначена для указания ресурсов (библиотек DLL и управляемого кода), которые обрабатывают определенные типы запросов.

Сгруппировать по: Состояние

Имя	Путь	Состояние	Тип ...	Обработчик
Отключен				
CGI-exe	*.exe	Отключен	Файл	CgiModule
ISAPI-dll	*.dll	Отключен	Файл	IsapiModule
Включен				
ASPClassic	*.asp	Включен	Файл	IsapiModule
AssemblyResourceLoader-Integr...	WebReso...	Включен	Не ...	System.Web.Ha
AXD-ISAPI-2.0				IsapiModule
CrystalImageHan				CrystalDecisions.We
HttpRemotingHan				System.Runtime.Rer
HttpRemotingHan				IsapiModule

Изменение разрешений функции

Разрешения:

- Чтение
- Сценарий
- Выполнение

Действия

- Добавить управляемый обработчик...
- Добавить сопоставление сценария...
- Добавление сопоставления сценария с подстановочными знаками...
- Добавить сопоставление модуля...
- Изменить...
- Удалить**
- Изменение разрешений функции...
- Вернуться к наследованным
- Справка в Интернете

Включить модуль **IsapiModule** (файл isapi.dll) для подключения собственных обработчиков **.dll**

Включить разрешение выполнения для этого модуля

(В IIS 5, 6 – это «Разрешён запуск» - «Сценарии и исполняемые файлы»)

Шаг 3 – добавить ISAPI-обработчик – «Добавить сопоставление сценария»

Подключения

- Начальная страница
- PPLDONMU (PPLDONMU)
- Пулы приложений
- Узлы
 - Default Web Site
 - App_Data
 - aspnet_client
 - Code backup
 - people
 - phys
 - router
 - script**
 - services

Сопоставление сценария

Эта функция предназначена для обработки определенных запросов.

Сгруппировать по: Сценарий

Включен

- ASPClassic
- AssemblyResourceLoader
- AXD-ISAPI-2.0
- CGI-exe
- HttpRemotingHandlerFactory
- HttpRemotingHandlerFactory
- HttpRemotingHandlerFactory
- HttpRemotingHandlerFactory
- ISAPI-dll

Добавление сопоставления сценария

Путь запроса: .dll

Пример: *.bas, wsvc.axd

Исполняемый файл: C:\inetpub\wwwroot\script\test.dll

Имя: test

Ограничения запроса...

Ограничения запроса

Сопоставление | Команды | Доступ

Укажите необходимый для обработчика доступ:

- Нет
- Чтение
- Запись
- Сценарий
- Выполнение**

Действия

- Добавить управляемый обработчик...
- Добавить сопоставление сценария...**
- Добавление сопоставления сценария с подстановочными знаками...
- Добавить сопоставление модуля...
- Изменить...
- Переименовать
- Удалить**
- Изменение разрешений

Итог – запись в «Ограничениях ISAPI и CGI» компьютера и фиксация настроек приложения в его файле конфигурации web.config

Подключения

- Начальная страница
- PPLDONNU (PPLDONNU)**
- Пульты приложений
- Узлы
 - Default Web Site
 - App_Data
 - aspnet_client
 - Code backup
 - people
 - phys
 - router
 - script
 - services

Ограничения ISAPI и CGI

Эта функция предназначена для указания расширений ISAPI и CGI, которые могут выполняться на веб-сервере.

Сгруппировать по: Без группирования

Описание	Ограничение	Путь
[Нет описания]	Разрешено	C:\inetpub\wwwroot\script\test.dll
Active Server Pa...	Разрешено	%windir%\system32\inetsrv\asp.dll
ASP.NET v2.0.5...	Разрешено	%windir%\Microsoft.NET\Framework\v2.0.50727\aspnet_isa...

Можно проверить добавление обработчика в ограничениях ISAPI и CGI вашего компьютера

```
web.config - Блокнот
Файл  Правка  Формат  Вид  Справка
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.webServer>
    <handlers accessPolicy="Read, Execute, Script">
      <add name="test" path=".dll" verb="*" modules="IsapiModule"
        scriptProcessor="C:\inetpub\wwwroot\script\test.dll"
        resourceType="Unspecified" />
    </handlers>
  </system.webServer>
</configuration>
```

ИСТОЧНИКИ

1. **Эспозито Д.** Microsoft ASP.NET 2.0. Базовый курс / Пер. с англ. – М.: «Русская редакция»; СПб.: Питер, 2007.-688 с.: ил.
2. **Эспозито Д.** Microsoft ASP.NET 2.0. Углублённое изучение / Пер. с англ. – М.: «Русская редакция»; СПб.: Питер, 2007.-592 с.: ил.
3. Основы архитектуры IIS, или запросопровод для ASP.NET. — <http://habrahabr.ru/post/189086/#par1>
4. Библиотека MSDN. — <http://msdn.microsoft.com/ru-ru/>