

Knowledge Base Article: Article 234

Web Portal Timeouts

Date: February 28th, 2012

Description:

Web portal uses http or https protocols which are not connected protocols, IIS and ASP.NET have mechanisms to handle end user sessions like idle timeout and session timeout.

It is also important to memorize end user selections between two sessions as language, skin... In that case cookies are used to store information on the end user device.

Further, Web Portal needs to communicate information to Web Builder or Web Dashboard when they start. In that case temporary cookies are also used.

1. Cookies timeouts:

In web.config, located in C:\inetpub\wwwroot\dwweb, you will find:

```
<add key="CookieTimeoutNav" value="90"/>  
<add key="CookieTimeoutIDs" value="5"/>
```

`CookieTimeoutNav` is used to store end user selection choices as language, skin... between sessions.

Default value is 90 days.

`CookieTimeoutIDs` is used to store IDs that have to be communicated between Web Portal and Web Builder or Web Dashboard. It is also used for internal authentication mechanisms.

Default value is 5 minutes.

2. Execution timeout:

If the server takes more than `executionTimeout` to finish the required treatment, the web browser will stop waiting for it and release the user from this waiting.

In web.config, you will find:

```
<httpRuntime executionTimeout="300"/>
```

Default value is 300 seconds (5 minutes).

WARNING: this value should not be bigger than the idle timeout or the session timeout described just after.

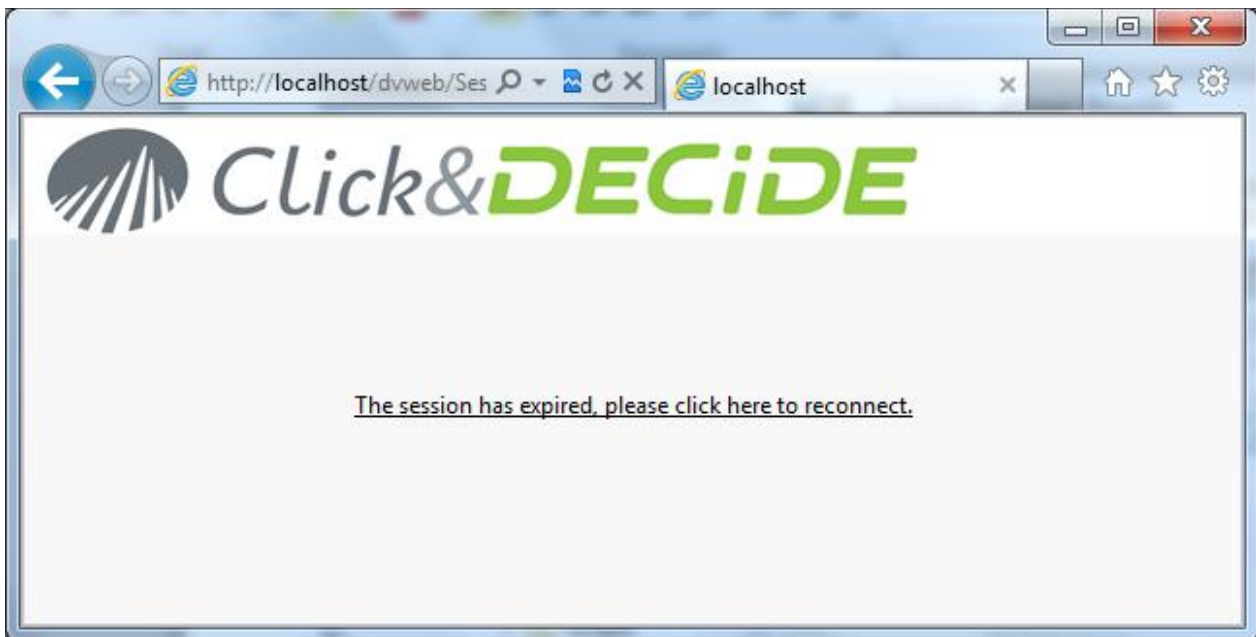
3. Session timeout:

In IIS, they are 2 settings:

- *Idle timeout*: if there is no activity during this period, the session is closed (20 minutes by default)
- *ASP.NET Session timeout*: after this period, the session is closed (20 minutes by default) to release server resources

WARNING: *ASP.NET Session timeout* should not be greater than *Idle timeout*.

Once a session is closed, Click&DECiDE displays a message and redirects the end user to the home page.

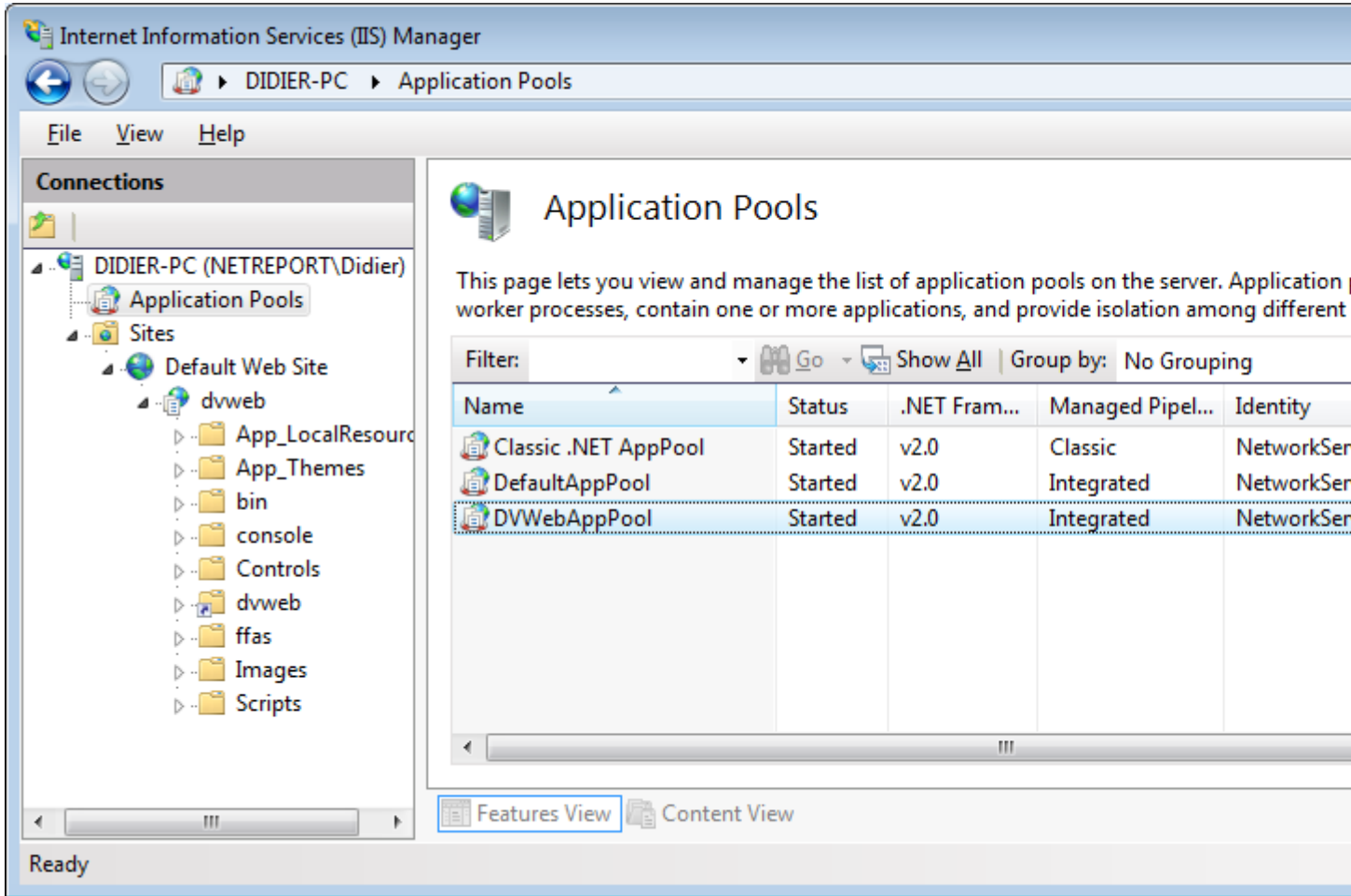


In Windows authentication mode, the authentication can be transparent.

In Click&DECiDE authentication mode, the user will have to supply its credentials again.

Idle timeout

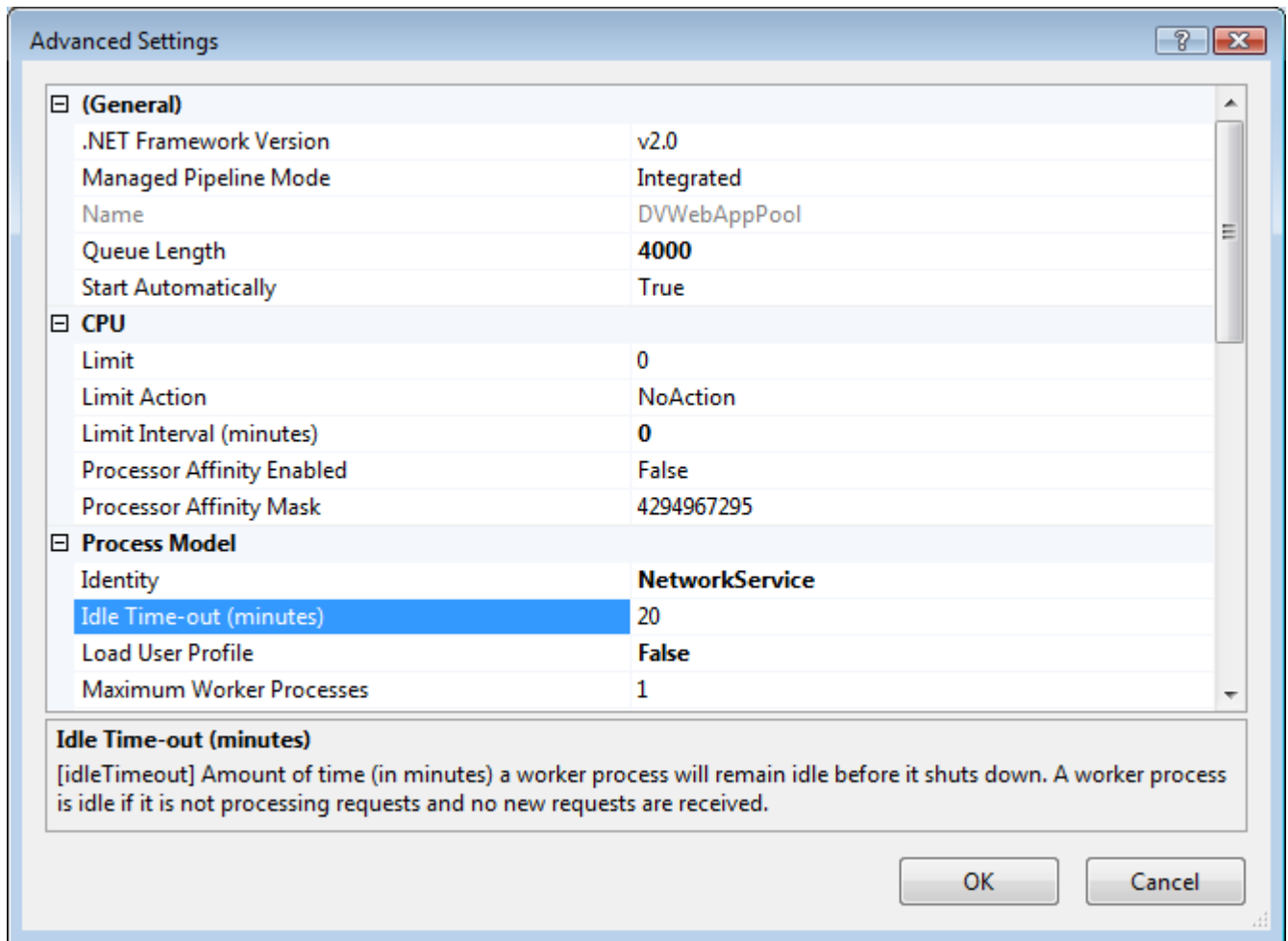
From IIS Services Manager, select *Application Pools*, then *DVWebAppPool*



The screenshot shows the IIS Manager console with the 'Application Pools' page selected. The left-hand tree view shows the hierarchy: DIDIER-PC (NETREPORT\Didier) > Application Pools > Sites > Default Web Site > dvweb. The main pane displays a table of application pools.

Name	Status	.NET Fram...	Managed Pipel...	Identity
Classic .NET AppPool	Started	v2.0	Classic	NetworkSer
DefaultAppPool	Started	v2.0	Integrated	NetworkSer
DVWebAppPool	Started	v2.0	Integrated	NetworkSer

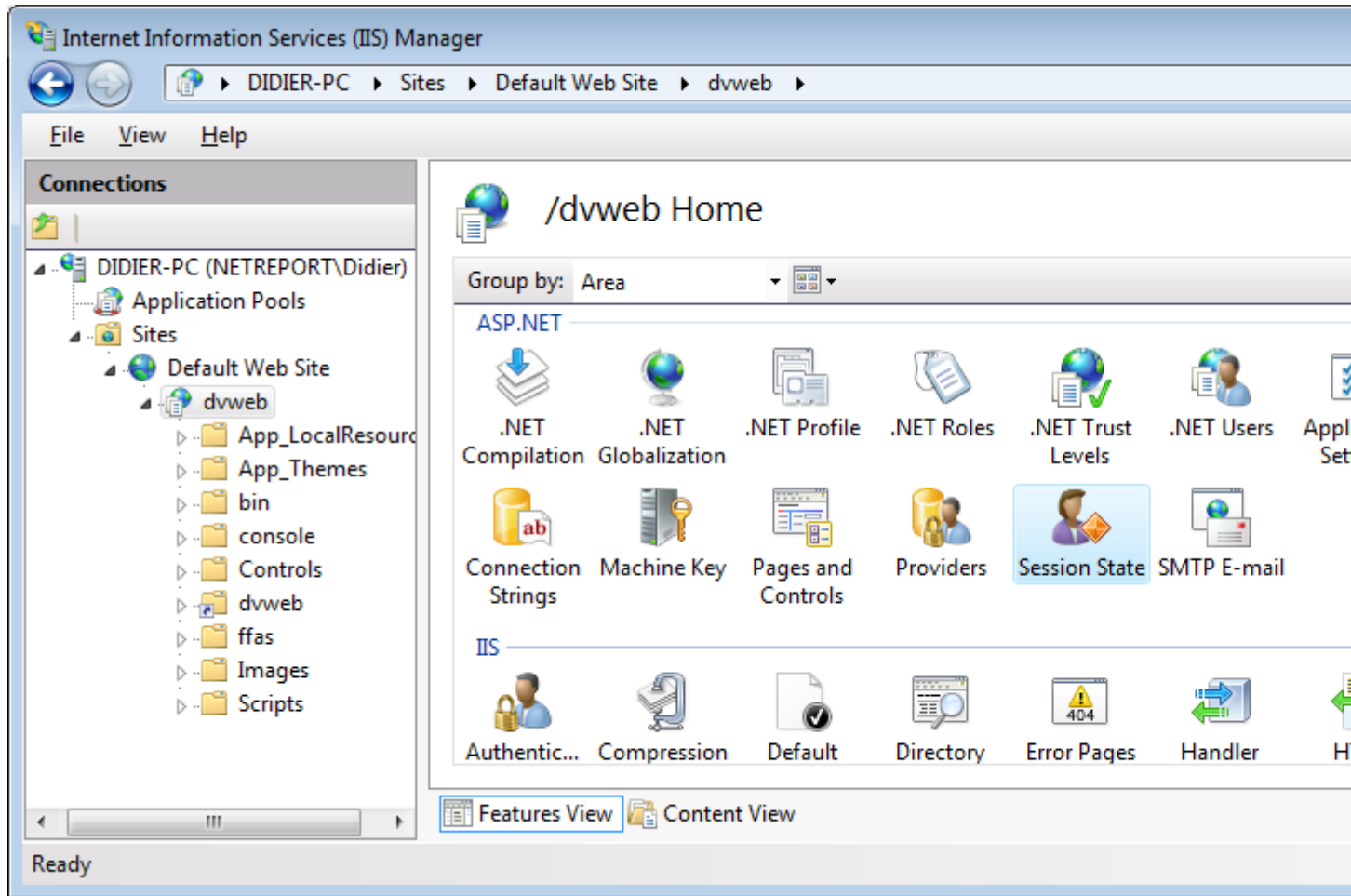
Right click on *DVWebAppPool*, then *Advanced Settings*.



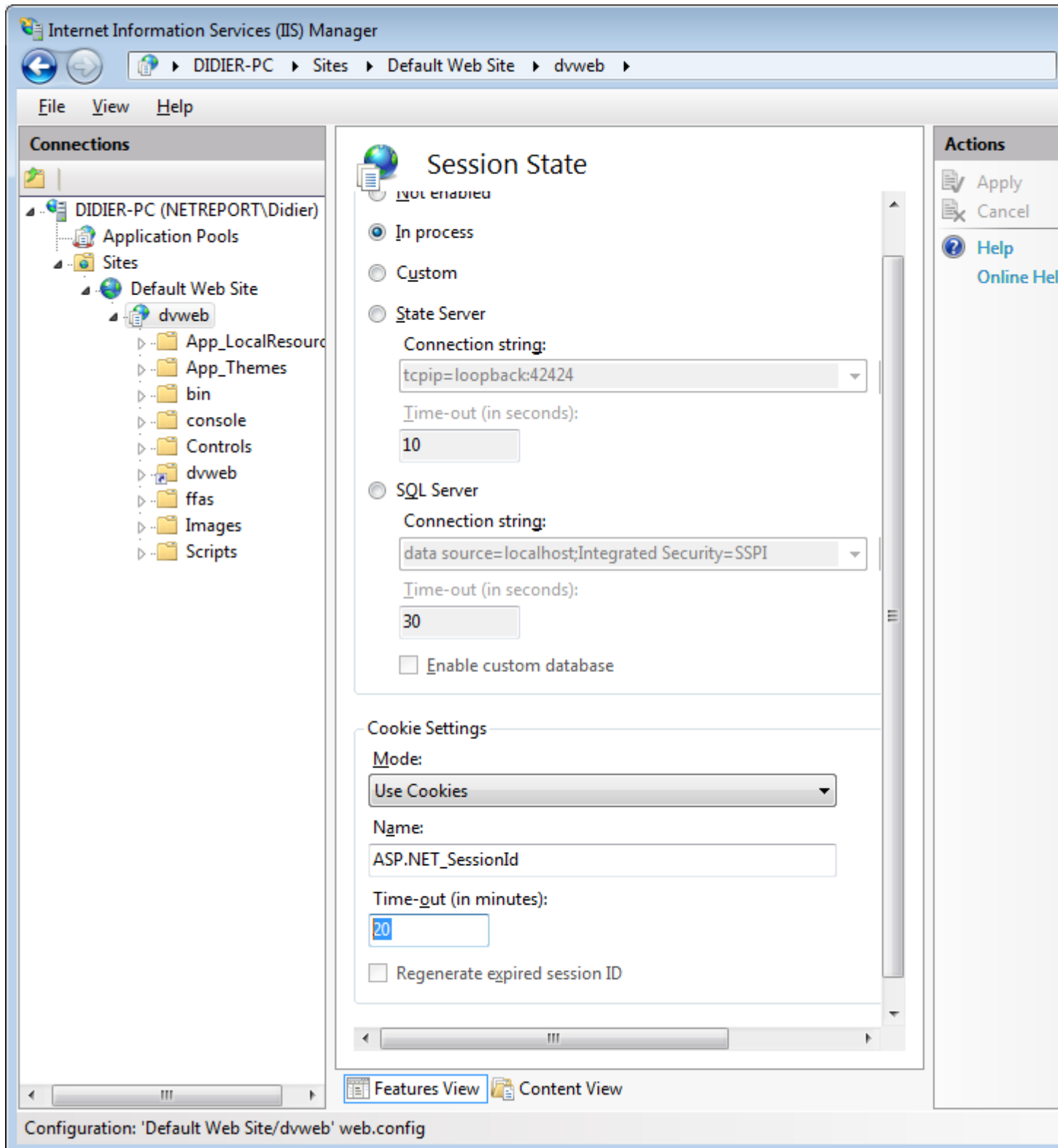
Adjust *Idle Time-out (minutes)* as you want.

Session Timeout

From IIS Services Manager, select *Default Web Site*, then *dvweb*, then double click on *Session State*



The screen below should appear



The screenshot shows the Internet Information Services (IIS) Manager interface. The left-hand pane displays the 'Connections' tree, with the path expanded to 'DIDIER-PC (NETREPORT\Didier) > Sites > Default Web Site > dvweb'. The main pane is titled 'Session State' and shows the following configuration:

- NOT enabled
- In process
- Custom
- State Server
 - Connection string: tcpip=loopback:42424
 - Time-out (in seconds): 10
- SQL Server
 - Connection string: data source=localhost;Integrated Security=SSPI
 - Time-out (in seconds): 30
 - Enable custom database

Below the Session State section, the 'Cookie Settings' section is visible:

- Mode: Use Cookies
- Name: ASP.NET_SessionId
- Time-out (in minutes): 20
- Regenerate expired session ID

At the bottom of the main pane, there are buttons for 'Features View' and 'Content View'. The status bar at the bottom indicates the configuration file: 'Configuration: 'Default Web Site/dvweb' web.config'.

Adjust *Time-out (minutes)* as you want.