

# Troubleshoot IIS Web Server Configuration in Enterprise Chat and Email

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Web Server Architecture](#)

[App Server Interactions](#)

[App Server Configuration](#)

[HTTP Response Headers in IIS](#)

[IIS Manager Authorization](#)

## Introduction

This document describes ECE Web server design and shows how to solve some of the most common problems related to IIS.

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

- Enterprise Chat and Email (ECE)
- Microsoft Internet Information Services (IIS)

### Components Used

The information in this document is based on these software versions:

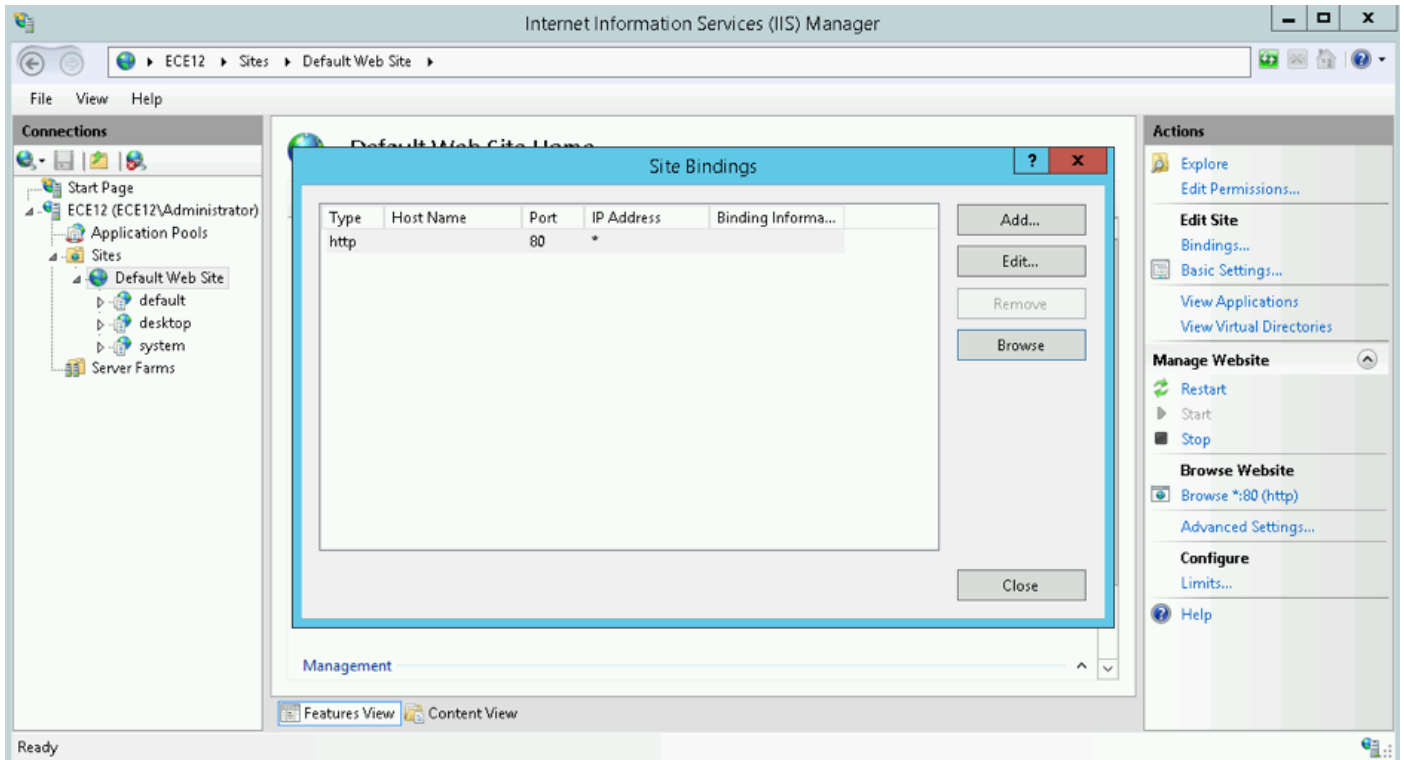
- ECE 11.6, 12.0

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration.

## Web Server Architecture

ECE 11.6 and 12.0 is based on IIS version 8.5. ECE Web server uses IIS to work as a front-end server, ECE Application Server works as a Backend server. The ECE Web server's main job is to display the site content (default folder **C:\ECE\Service**) and the application server is in charge of the logic, the interaction between the end-user and the displayed content. The ECE application server works in conjunction with the web server, where one displays and the other one interacts.

All incoming traffic goes to front-end. This means that it needs to listen to port 80/443, based on the site bindings (**IIS Manager > Default Web Site > Bindings**)



**Note:** This is the default view when communication with the WEB server is only through HTTP. Detailed instructions on how to activate HTTPS can be found in the **ECE Installation and Configuration Guide**, chapter **SSL Configuration**.

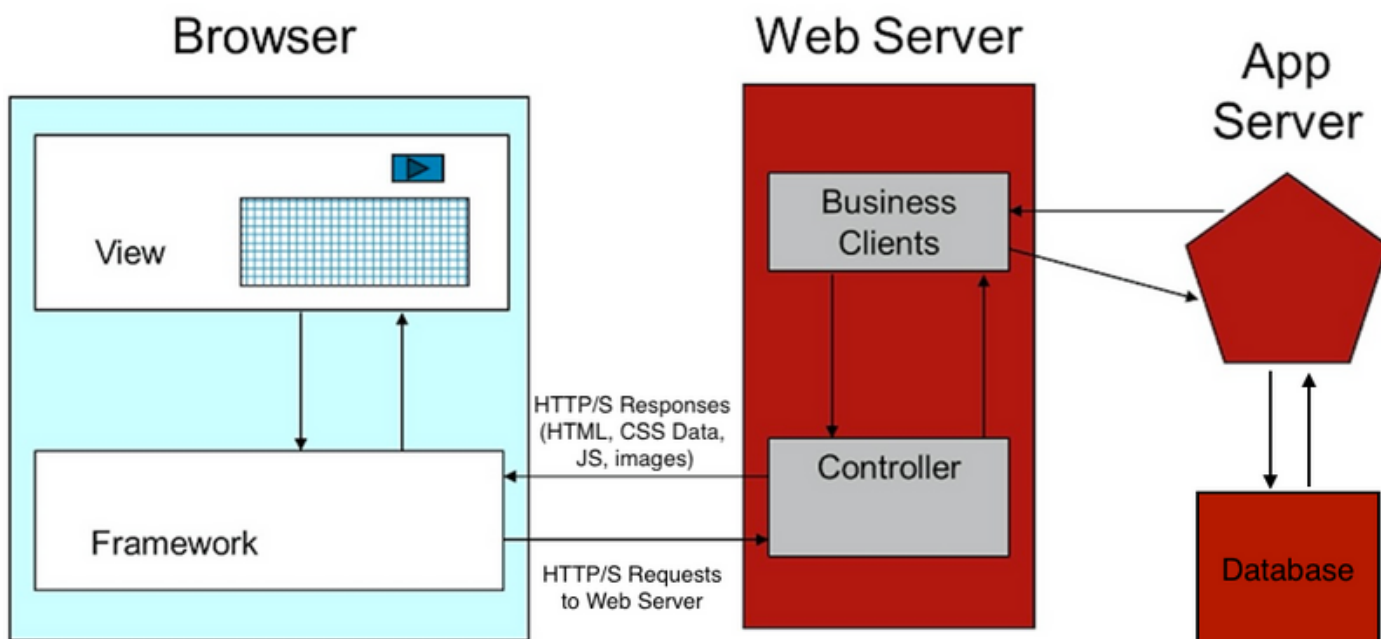
HTTP works as a request-response protocol between a client and a server, where the client is a web browser. Everything starts when the browser submits an HTTP request to the server; that request is processed by the ECE Web server, get the response from the ECE Database and returns as an HTTP response to the client.

The basic HTTP data flow here is **Web Browser > ECE Web Server (IIS) > ECE Application Server > ECE Database Server (MS SQL)**. The web servers can be configured behind the load balancer.

**Tip:** In ECE environment we suggest to keep ECE Web server and/or LB in the DMZ.

# The User Interface Architecture

## Data Flow



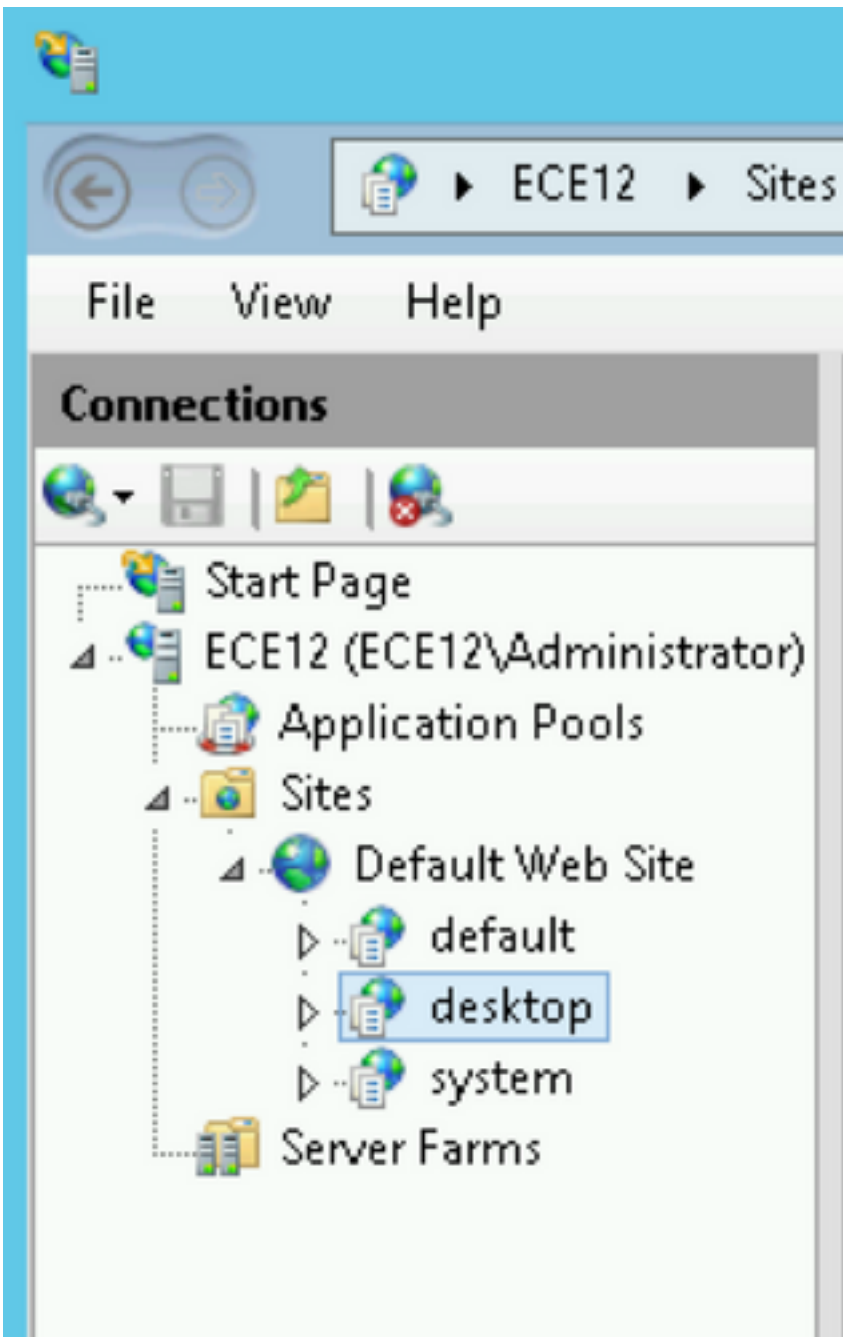
## App Server Interactions

How Web server redirect the request to the APP server?

ApplicationHost.config is the root file of the IIS configuration. It includes definitions of all sites, applications, virtual directories and application pools, as well as global defaults for the web server settings.

The ApplicationHost.config file can be found in the folder `%WINDIR%\system32\inetsrv\config`

When ECE Web server component is installed, the installation process makes changes, based on ECE requirements to the core files of IIS, which is ApplicationHost.config. As part of the installation, it creates two applications. The first one is the system application which has all the content related to partition 0. And the default application which relates to partition 1. That includes the admin console, light agent, supervisor console, reports and etc. In ECE 12.x the light agent is created as a part of a separate application and in 12.x customers can use it in production without Finesse.



These settings are added by ECE installer to ApplicationHost.config for the applications:

```

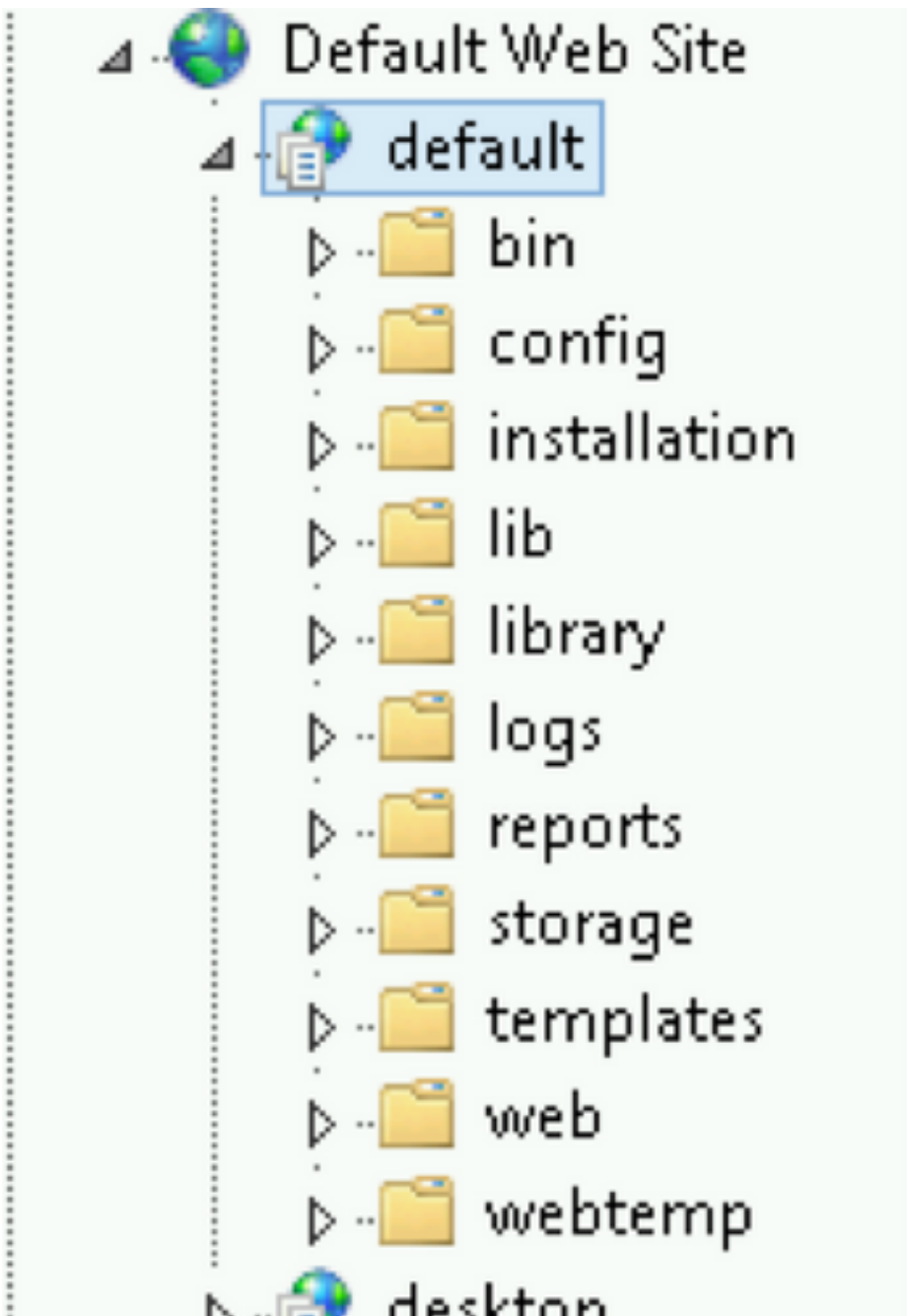
<sites>
  <site name="Default Web Site" id="1">
    <application path="/">
      <virtualDirectory path="/" physicalPath="%SystemDrive%\inetpub\wwwroot" />
    </application>
    <application path="/system" applicationPool="DefaultAppPool">
      <virtualDirectory path="/" physicalPath="C:\ECE\eService" />
      <virtualDirectoryDefaults
        userName="CC\ipccadmin"password="[enc:AesProvider:hzS/wGBY0K6+gZcYgQstzLqbst1lv71LMBwdPyPZyUM476
        9Txc4kCGpDgX3xc96L:enc]" />
    </application>
    <application path="/default" applicationPool="DefaultAppPool">
      <virtualDirectory path="/" physicalPath="C:\ECE\eService" />
      <virtualDirectoryDefaults userName="CC\ipccadmin"
        password="[enc:AesProvider:0LYLPJwWZjuKhRHe2WLBalYsk8Zj8hiPqUbqzKVRKKI=:enc]" />
    </application>
    <application path="/desktop" applicationPool="DefaultAppPool">

```

```
<virtualDirectory path="/" physicalPath="C:\ECE\eService" />
<virtualDirectoryDefaults userName="CC\ipccadmin"
password="[enc:AesProvider:VtxnHOST1S2xo1er82ROFEAyC4PeF5a9qmasmaR+W0g=:enc]" />
</application>
<bindings>
  <binding protocol="http" bindingInformation="*:80:" />
  <binding protocol="https" bindingInformation="*:443:" sslFlags="0" />
</bindings>
</site>
```

The ECE installer takes care about everything, you must not do any manual configuration changes in the IIS Management console. If there are no changes, everything works as expected.

This is the entire layout of **eService** folder that is available under the installation directory **C:\ECE\eService**



The system gives access to the entire eService folder because there are multiple components that must be reachable from IIS for correct application's functionality.

- **bin** contains all the initialization/startup files

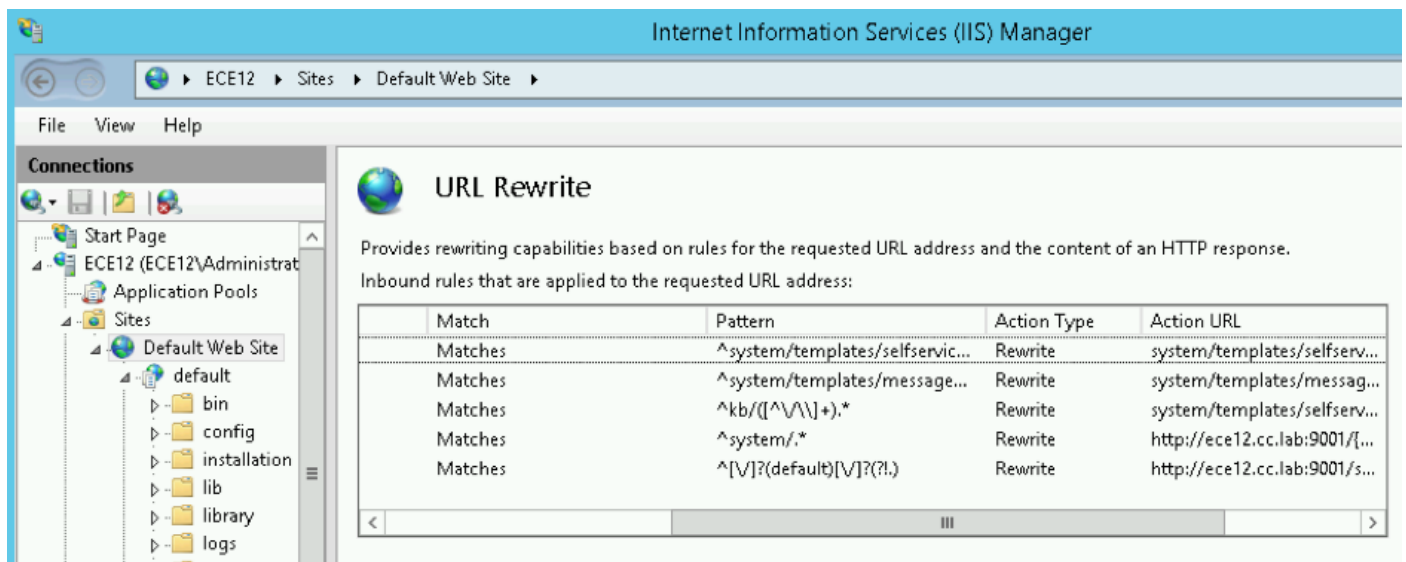
- **config** contains all configuration files
- **log** contains the real-time logs of the application

The primary folders which are used by IIS are **templates** and **web**.

- **templates** used for every request which comes to entry point (chat/callback template)
- **web** used for any requests to web console (admin, system, tools, etc)

Any other requests from the platform are handled by redirection. For example, when you use agent availability API request

(**ECE\_webserver\_name/system/egain/chat/entrypoint/agentAvilability/1000**) the IIS uses the redirect URL rewrite rules to redirect that request to the ECE application server port 9001:



Depends on ECE version, the rewrite rules are stored either in the **ApplicationHost.config** file or in **Web.config** files which are located in **%SystemDrive%\inetpub\wwwroot**. In this file, you can see the **<rewrite>** section that contains this rule definition.

The installer takes care of the rewrite section, but sometimes this data can be modified and as a result, the chat templates or APIs don't work properly. So that is the first place which must be checked when you see such symptoms.

## App Server Configuration

In a non-redundant solution, the application server does not require any customization. The installer pre-deploys and configures the system by himself. But in High Availability (HA) deployment, ECE can be installed with multiple WEB and application servers. In such case for each web server in the deployment must be installed a separate application server (1 to 1 map).

How do you identify which application server is associated with which web server?

The quickest way to check this is to open the configuration file which is located on the web server and check the **worker.default.host** parameter:

**C:\ECE\service\installation\egain-plugin\workers.properties**

```

ps=\
worker.maintain=60

# An entry that lists all the workers defined
# 'default' worker -- Will be used for jsp's
# 'pushlet' worker -- Will be used for Pushlet requests [nailed connection]
# 'live' worker -- Will be used for live requests
# 'forum' worker -- will be used for community/forum requests

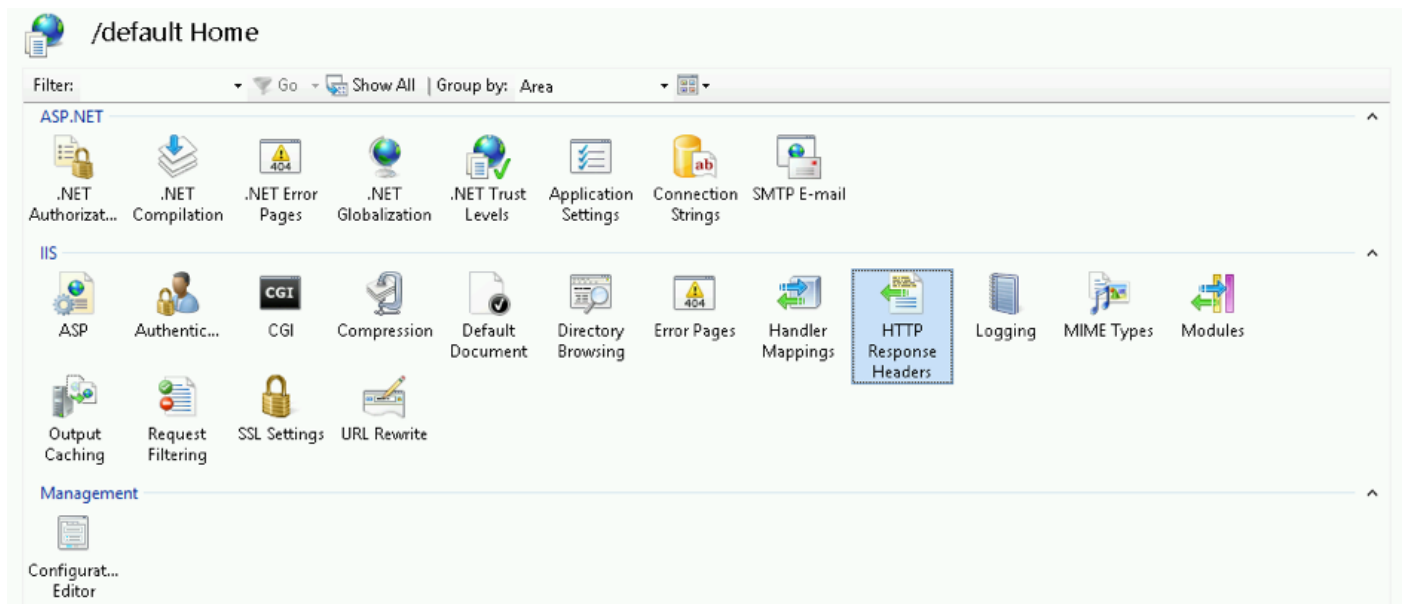
worker.list=default

# Entries that define the host and port associated with these workers
worker.default.host=ece.cc.lab
worker.default.port=15006
worker.default.type=ajp13
worker.default.ping_timeout := 30000
worker.default.ping_mode := CI
worker.default.socket_timeout := 3600
worker.default.connection_pool_timeout := 3600
worker.default.socket_keepalive := true
worker.default.connection_pool_size := 2000
worker.default.retry_interval := 150

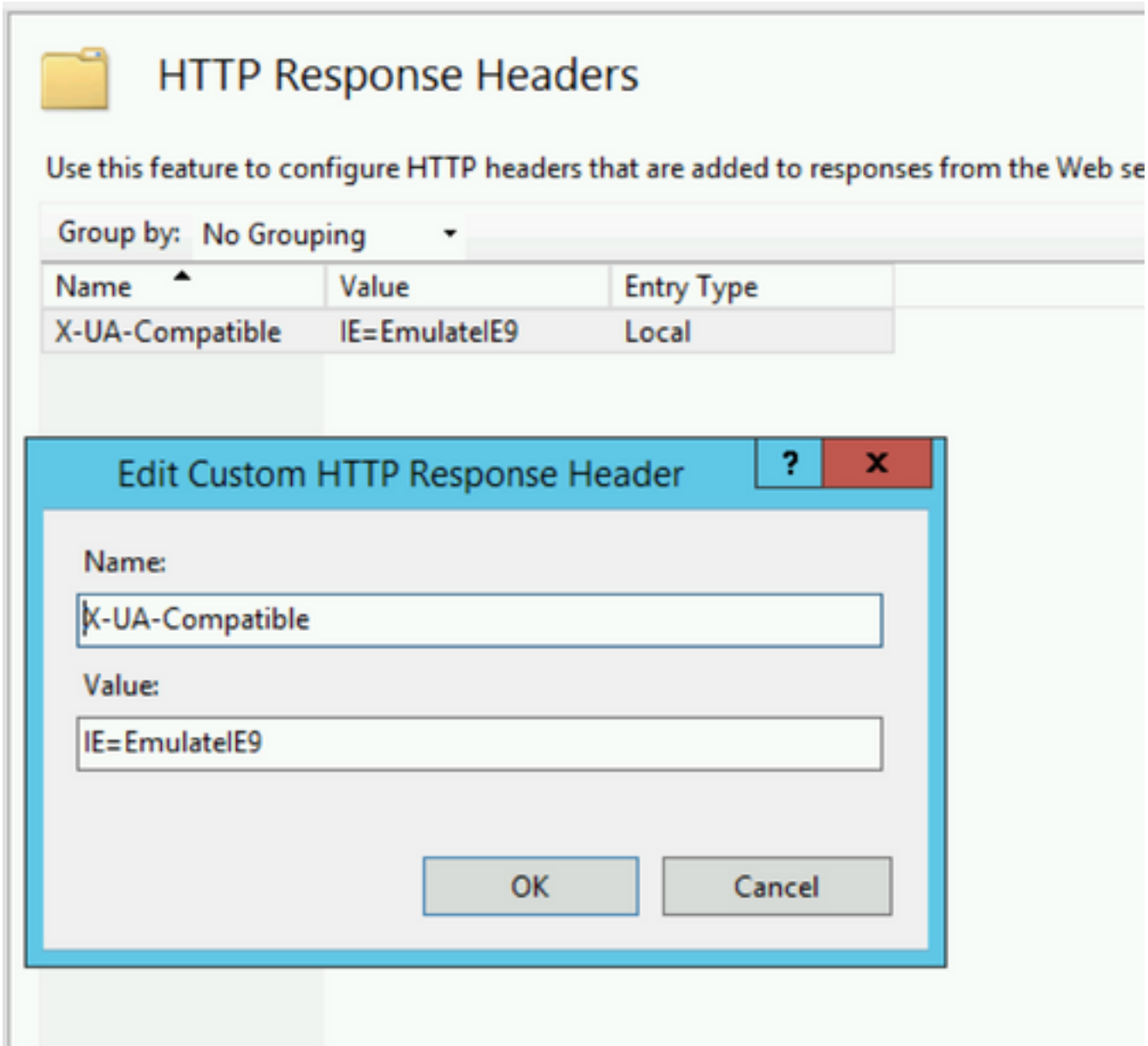
```

## HTTP Response Headers in IIS

HTTP Response headers are name-value pairs of strings sent back from a server with the content you requested. They are typically used to transfer technical information like how a browser must cache content, what type of content it is, the software run on the server and more. Increasingly, HTTP response Headers are the one who validated by the client to render responses coming in from the server. It can be a static request or dynamic requests.



In ECE every application has it's unique HTTP Responses Headers properties. Any requests which come to the web server has this header by the default:



**X-UA-Compatible** is a document mode meta tag that allows choosing what version of Internet Explorer the page must be rendered as. It forces the browser to render as that particular version's standards (as of IE 9).

In some cases the installer don't add **X-UA-Compatible** header during the installation. That can happen due to the account's permissions which used during the installation. As a workaround, this parameter can be added manually.

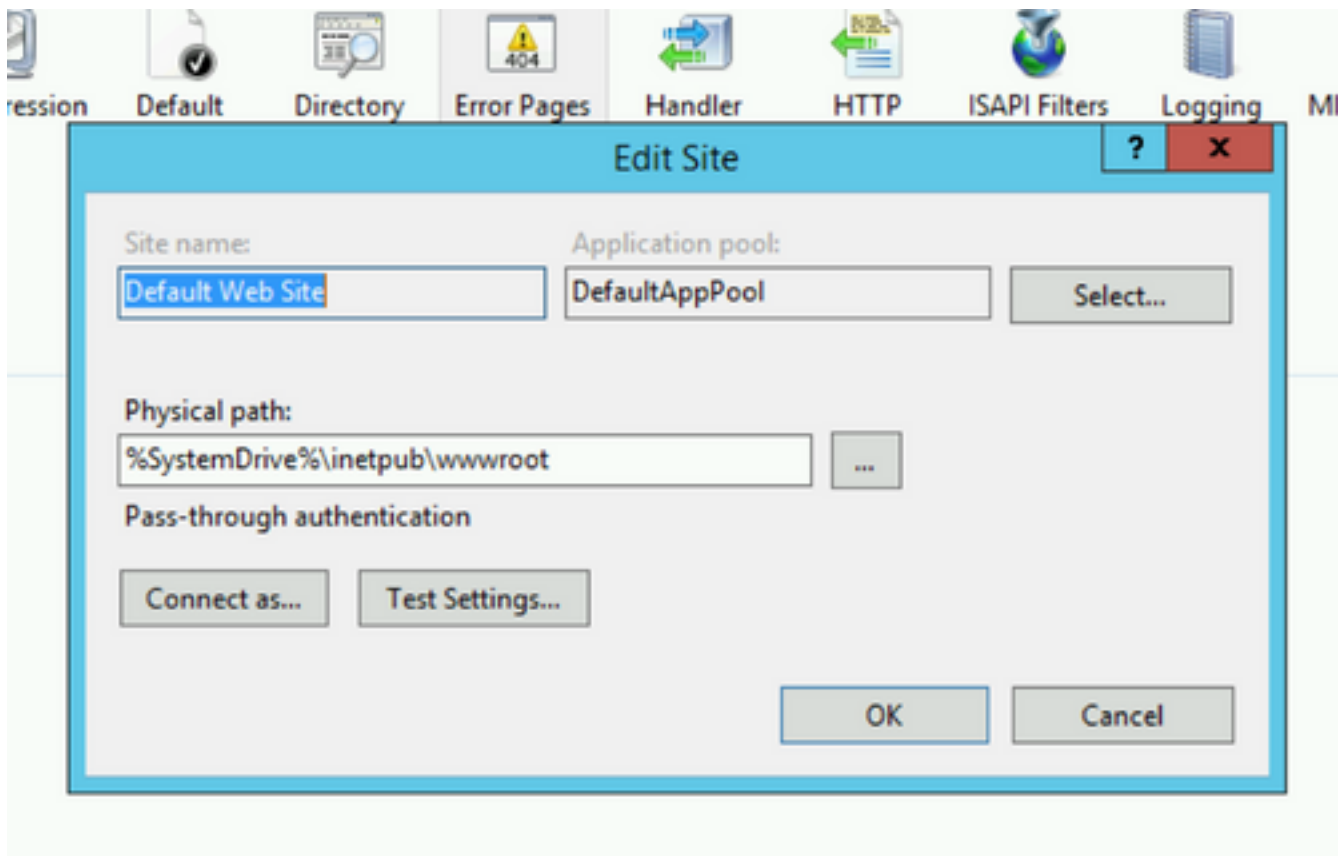
**Caution:** There must not be any other headers.

## IIS Manager Authorization

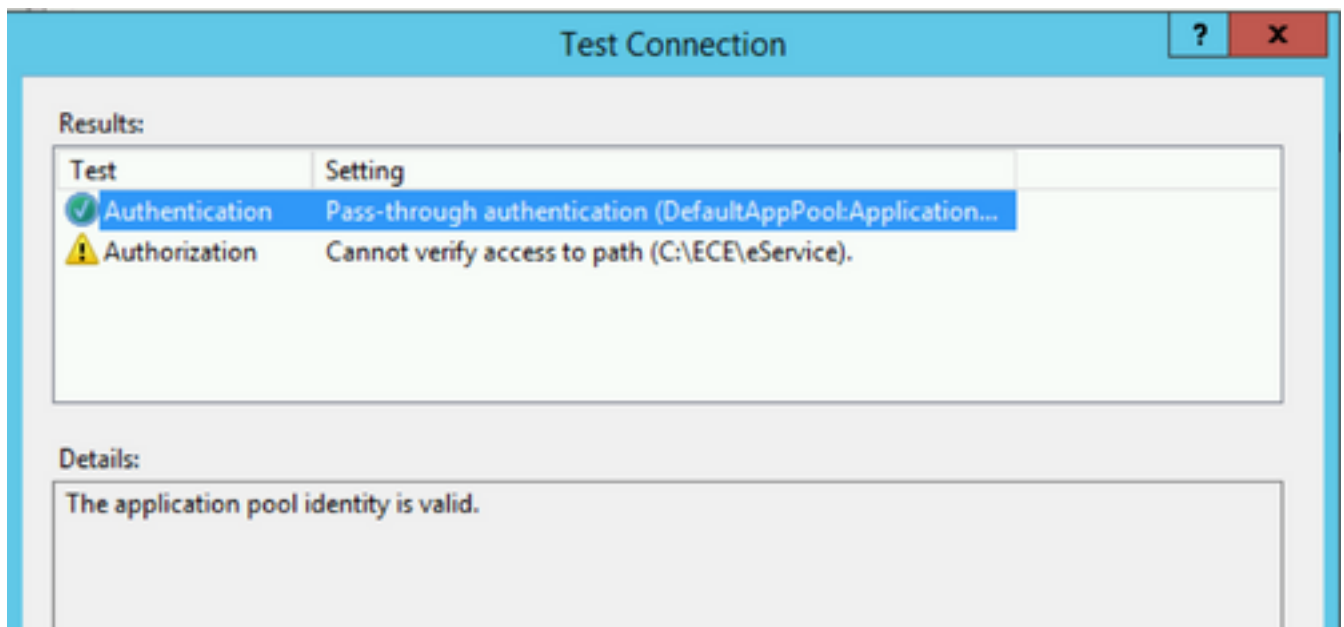
In the case when the application server is up and running, but you are not able to go through the application server and the IIS page gives an error, it can be helpful to check the IIS Authorization settings. It can be that the account which was used during installation is not able to read the **eService** folder.

If you select your Application pool in **IIS Manager** and choose the **basic settings**, you see this window:





Here the installer specifies the username and password who has complete privileges to perform operations on the default path. Ensure that the application pool identity has Read access to the physical path by select **Test Settings...** button. If you see a yellow icon next to **Authentication** or **Authorization** you must check the account's credentials and permissions.



Also, it can be a scenario when you move the WEB server from the local network to DMZ. In case if the current domain user is not existing in the DMZ, the new user credentials must be updated here as well. It can be either a domain user or a local admin.