SmartSpace®

Scaling out SmartSpace Web

From version 3.8

Ubisense Limited, St Andrew's House, St Andrew's Road, Cambridge CB4 1DL, United Kingdom.

Telephone: +44 (0)1223 535170. Website: https://www.ubisense.com

Copyright © 2023, Ubisense Limited 2014 - 2023. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Ubisense at the following address:

Ubisense Limited
St Andrew's House
St Andrew's Road
Cambridge CB4 1DL
United Kingdom

Tel: +44 (0)1223 535170

WWW: *https://www.ubisense.com*

All contents of this document are subject to change without notice and do not represent a commitment on the part of Ubisense. Reasonable effort is made to ensure the accuracy of the information contained in the document. However, due to on-going product improvements and revisions, Ubisense and its subsidiaries do not warrant the accuracy of this information and cannot accept responsibility for errors or omissions that may be contained in this document.

Information in this document is provided in connection with Ubisense products. No license, express or implied to any intellectual property rights is granted by this document.

Ubisense encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.

UBISENSE®, the Ubisense motif, SmartSpace® and AngleID® are registered trademarks of Ubisense Ltd. DIMENSION4™ and UB-Tag™ are trademarks of Ubisense Ltd.

Windows® is a registered trademark of Microsoft Corporation in the United States and/or other countries. The other names of actual companies and products mentioned herein are the trademarks of their respective owners.

Contents

Introduction to Scaling out SmartSpace Web ... ***1***

Requirements ... *1*

The Web Partition ... ***2***

Configuring a Web Partition ... ***3***

Website ... *3*

Windows ... *3*

Linux ... *3*

Forwarded headers ... *4*

Services ... *4*

Windows ... *4*

Linux ... *5*

Client Interface Services Required on a Controller ... *5*

Load balancing and reverse proxy ... *5*

Windows ... *6*

Linux ... *6*

Session Stickiness ... *6*

Reports Engine and SQL Scaling ... ***7***

Introduction to Scaling out SmartSpace Web

This guide describes the support within SmartSpace for scaling out the SmartSpace website. Here scaling out means supporting multiple back-end web servers so that a load balancer can spread load across those servers. This enables you to support more users by adding web servers.

This is an advanced subject, and some knowledge of website hosting and load balancing technologies is assumed. You should also be familiar with the general SmartSpace website installation methods described in SmartSpace Installation.

Requirements

- SmartSpace version 3.7 or higher
- ASP.NET Core Runtime 6.0.x Hosting bundle for Windows or ASP.NET Core Runtime 6.0.x for Linux

The Web Partition

The basic support for web partitions was introduced in SmartSpace 3.7. A partition allows a given host server to run an isolated copy of the SmartSpace website, along with the services that process queries from users.

Configuring a Web Partition

A web partition is configured by setting the machine-local configuration parameter **web_partition** to be a string value that is unique across machines in your platform. For example, it might be the IP address of the local machine, or another unique label such as a partition number.

Choose a partition name – here represented by "XXXXXXXX".

On a Windows host, add the following value to the registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Ubisense 2.1\Platform\Config\web_partition
    REG_SZ
    XXXXXXXX
```

On a Linux host, add the following to **/etc/ubisense/platform.conf**:

```
web_partition: XXXXXXXX
```

After the parameter is set, any instances of the SmartSpace website run on the host will attempt to connect to a partition-specific copy of the services used to run web map, forms, and reporting queries. Furthermore, any instances of those services will advertise the partition-specific versions of their schemas. The next step is to configure local versions of these to be started.

Website

With the partition set, a copy of the website is run on the local machine. This copy will attempt to connect to a partition-specific version of the client interface services.

Windows

For the Windows website, install the website from the msi, as in the normal installation instructions.

Linux

On Linux, it is suggested that the website for a partition should be run using systemd. To do this, first extract the website from the distributed file **web/linux/packages/i586_linux_2.6/ubisense_smartspace_web.zip** to a folder, such as **/home/platform/website**. Then set up a systemd script to start the service and restart if it exits. For example, the following could be placed in **/etc/systemd/system/website.service**:

```
[Unit]
Description=Ubisense Website
After=network.target
StartLimitInterval=0

[Service]
Type=simple
Restart=always
RestartSec=1
User=platform
WorkingDirectory=/home/platform/website
ExecStart=/home/platform/website/ubisense_smartspace_web

[Install]
WantedBy=multi-user.target
```

Then run **systemctl start website** and **systemctl enable website**.

Forwarded headers

If using a reverse proxy on another host, such as a load balancer, the website must also be configured to use standard forwarded headers from that proxy. To do this, add the "AcceptHeadersFrom" configuration to the ProxyOptions section in the **localsettings.json** file if using Windows or **web.json** on Linux.

```
"ProxyOptions": {
"Base": "/SmartSpace",
"AcceptHeadersFrom": "10.42.1.32"
}
```

Once this is set, the website will expect to receive the request protocol in "X-Forwarded-Proto", and the requesting client address in "X-Forwarded-For". These allow the website code to handle authentication requirements correctly. If these are not set then the website may for example respond with a redirect to https if it believes that a secured connection is required by authentication, resulting in a possible redirect loop.

Services

With the partition set, copies of the client interface web services must be started to advertise and serve the partition schemas.

Windows

Ubisense suggest that a tool such as NSSM could be used to start and restart instances of the following executables (from the **packages/i586_windows_1.3** folder of the distribution):

- **ubisense_client_interface.exe**
- **ubisense_ai_client_interface_oracle.exe** *or* **ubisense_ai_client_interface_sqlserver.exe**
- **ubisense_assoc_and_config_web_interface.exe**
- **ubisense_object_view_web_interface.exe**

Linux

Again, use systemd to start instances of the following executables (from the **packages/i586_linux_2.6** folder of the distribution):

- **ubisense_client_interface**
- **ubisense_ai_client_interface_oracle** or **ubisense_ai_client_interface_sqlserver**
- **ubisense_assoc_and_config_web_interface**
- **ubisense_object_view_web_interface**

Client Interface Services Required on a Controller

When installing website partitions, you should leave the map/forms/report client interface services still deployed on a controller (often the same machine the Core Server is running on, although any controller will do). Check where client interface services are deployed using Service Manager. The services to look for are:

- Web map client interface
- Web forms client interface
- Report client interface SQL Server or Report client interface Oracle
- Object view API client interface

Even though these client interface service instances will not be used directly, as each partition will use its own locally running, manually managed instances, they must still be deployed, otherwise the website will hide the map/form/reports menu items.

Load balancing and reverse proxy

In order to present a single website URL to users, you need to deploy some kind of load balancing reverse proxy. There are several possible choices here, such as Apache, NGINX, or managed solutions such as AWS Elastic Load Balancer. In all these cases, you need to ensure that the web service is accessible from the reverse proxy.

Windows

There are a number of load balancing options for IIS available. Within IIS, there is the Application Request Routing module (ARR). Normally SSL Offloading will be required in order to present a single SSL certificate.

Linux

By default, on Linux, the web service binds to URL 127.0.0.1:5000. This will not be accessible from a remote load balancer. To set the URL bound by the web service, pass an argument to the service. For example, using systemd, use the following in the **.service** configuration file:

```
  ExecStart=/home/platform/website/ubisense_smartspace_web --
urls=http://*:5000
```

Now set up a firewall rule to allow connections to 5000 only from the reverse proxy/load balancer.

The SSL should be handled at the load balancer, and the request scheme protocol passed to the web service as header X-Forwarded-Proto, so that it knows it is safe to use a login form and cookie authentication. Or authentication should be done at the reverse proxy and the authenticated user passed in the configured UserHeader. See the general SmartSpace website installation methods described in SmartSpace Installation.

Session Stickiness

Session persistence for creating an affinity between a client and a backend website must be set up.

Reports Engine and SQL Scaling

The partition configuration results in multiple copies of the reporting client interface service, but these will all be connecting to the same database server back-end, as configured in SmartSpace Config/PROPERTY HISTORY. The compilation and tracking of query execution will be federated, but the actual execution will not.

To scale the execution of report queries, it is suggested that the built-in cluster scaling methods of SQL Server or Oracle should be used, so that the single connection string results in queries being balanced across one or more instances of the database engine.