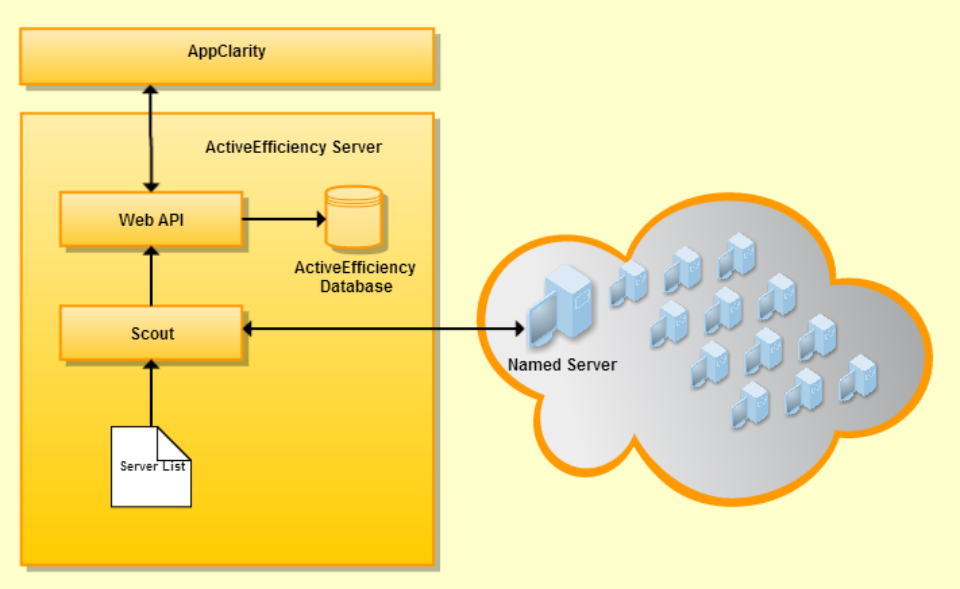
Configuring Server mode

This mode provides information on software installations for Windows and LINUX/UNIX servers to enable AppClarity to manage servers and server software.

The following diagram shows a schematic of the ActiveEfficiency components involved in the server support feature.



- The Web API provides an interface where server and server software information coming from the Scout can be stored and retrieved for use in AppClarity.
- The Scout scans a named list of servers and queries them for hardware and software inventory information.
- AppClarity is a consumer of the data captured from servers.

Windows Server Data Capture Requirements

To enable the Scout to remotely monitor Windows servers you will need to configure the following items on the target servers:

Scout for servers uses WMI and Remote Registry on Windows to discover information. Firewalls need to be configured to allow WMI connectivity, which consists of incoming traffic on TCP ports 135, 445 and additionally dynamically assigned ports, typically in the range of 1024 to 1034. These requirements must be met to enable the server Scout to remotely monitor servers for AppClarity:

To configure the firewall on a single computer using the computer's local settings on Windows Server 2008:

1. Log on to the target computer as the administrator.
2. From the Control Panel, open the Windows Firewall Settings dialog box.
3. On the Exceptions tab, check Windows Management Instrumentation (WMI).

To configure the firewall on a single computer using the computer's local settings on Windows Server 2008 R2 and above:

1. Log on to the target computer as the administrator.
2. From the Control Panel, locate and open the Windows Firewall configuration dialog.
3. Click Allow a program or feature through Windows Firewall.
4. Check Windows Management Instrumentation (WMI).

To configure the firewall on multiple domain computers using group policy where the windows firewall is configured to use the domain profile:

- Create a group policy object for the organizational unit that contains the Windows servers that you want to manage:
- Log on to a domain-member computer that is running Microsoft Windows Server 2008 and above. Log on with a user account that is a member of one or more of the following security groups:
    - Domain Admins
    - Enterprise Admins
    - Group Policy Creator Owners
- Click Start > Run, type mmc in the Open dialog box, and then click OK.
- On the File menu, click Add/Remove Snap-in.
- On the Standalone tab, click Add.
- In the Add Standalone Snap-in dialog box, click Group Policy Object Editor, and then click Add.
- In the Select Group Policy Object dialog box, click Browse.
- Click the group policy object that you want to update with the new Windows Firewall settings. For example, click the organizational unit that contains the Windows XP SP2 computers, click OK, and then click the group policy object that you created in step 1.
- Click OK, and then click Finish.
- Click Close, and then click OK.

- Under Console Root, expand the group policy object that you selected, and then click Computer Configuration > Administrative Templates > Network > Network Connections > Windows Firewall > Domain Profile.
- In the right pane, double-click Windows Firewall: Allow inbound remote administration exception.
- Click Enabled, and then specify the administrative scope in the Allow unsolicited incoming messages from dialog box. For example, to permit remote administration from a particular IP address, type that IP address in the Allow unsolicited incoming messages from dialog box.

  To permit remote administration from a particular subnet, type that subnet by using the Classless Internet Domain Routing (CIDR) format. In this scenario, type 192.168.1.0/24 to specify the network 192.168.1.0 with a 24-bit subnet mask of 255.255.255.0.

  For more information on how to specify a valid administrative scope, see the Syntax area of the Setting tab in this policy.
- Click OK, and then click Exit on the File menu.
- If the remote servers have the Windows firewall enabled you will need to configure each firewall for remote access.

    - Allow remote administration exception, or use a netsh firewall command at the command prompt to allow for remote administration on remote servers. The following command enables this feature:

```
netsh firewall set service RemoteAdmin enable
```

    - If you would rather use the Group Policy editor than the NETSH above command, use the following steps in the Group Policy editor (Gpedit.msc) to enable "Allow Remote Administration" on the servers:
        1. Under the Local Computer Policy heading, double-click Computer Configuration.
        2. Double-click Administrative Templates, Network, Network Connections, and then Windows Firewall.
        3. If the server is in the domain, then double-click Domain Profile; otherwise, double-click Standard Profile.
        4. Click Windows Firewall: Allow remote administration exception.
        5. On the Action menu, select Properties.
        6. Click Enable, and then click OK.

The following steps describe how to configure this using GPO:

1. On the domain controller select Start -> Administrative tools -> Group policy editor -> Edit an existing policy or add a new one
2. Navigate to Local Computer Policy -> Computer Configuration -> Policies -> Windows Settings -> Security Settings -> System Services
3. After reboot, all servers will have remote registry running.

- Windows Server 2012
- Windows Server 2008 R2
- Windows Server 2008 (32 and 64 bit)
- Windows Server 2003 SP1 or SP2 (32 and 64 bit) - You must have the following hotfix applied on 2003 platforms to get the information on number of cores and sockets:

    - Hotfix #932370: (WindowsServer2003.WindowsXP-KB932370-v3-x64-ENU.exe and WindowsServer2003-KB932370-v3-x86-ENU.exe, for more info:http://support.microsoft.com/kb/932370 - To apply this hotfix, you must have Windows Server 2003 Service Pack 1 (SP1) or Windows Server 2003 SP2 installed on the computer).

The Data Capture Account used to run the Scout must have specific DCOM and WMI permissions set.

LINUX or UNIX Server Data Capture Requirements

To enable the scout to remotely monitor LINUX or UNIX servers you will need to configure the following items on the target servers:

Scout for servers uses SSH on LINUX to discover information. The default SSH port is port 22. Keyboard interactive and password authentication are currently supported. The credentials for SSH login are held within a configuration file, credentials.config, found within the following directory:

%program files%\1E\ActiveEfficiency\Scout\Config

- RHEL 6.2 - 64 bit
- SLES 11 SP1 - 64 bit

Configuring AppClarity Integration

The ActiveEfficiency Connector and VirtualAppExporter components can all be configured to run on a schedule using Windows Task Scheduler.

The following table shows the recommended schedules for the components:

Component	Schedule
Scout	Once a day
ActiveEfficiency Connector	Synchronized using the AppClarity Connector Synchronizer once a day

We recommend that the components are also run in the order they appear in the table so that the latest information is always passed on immediately. If the Scout happens to run just after the ActiveEfficiency Connector has been synchronized the only side-effect will be that the data may, at worst, be out of date by 24 hours on a once a day schedule.

Running the Scout

In a default installation the scout is located in the following directory:

```
C:\Program Files (x86)\1E\ActiveEfficiency\Scout\
```

The scout in Server mode has two specific command-line parameters that allow you to scan Windows or Linux servers. To run the scout in just Server mode you use the `Modes` command-line parameter and set the value to `Server`, as described in Scout Command-line. When you run the scout in Server mode you will also need to add either the `Servers` or `NixServers` command-line parameters. The following example shows the scout run in Server mode only, and set to scan both UNIX/Linux and Windows servers:

```
C:\Program Files (x86)\1E\ActiveEfficiency\Scout\Scout.exe Modes=Server NixServers=nixserverlist.txt
Servers=winserverlist.txt
```

Scanning Windows servers

To scan Windows servers you use the `Servers` command-line parameter and run a command similar to the following:

```
C:\Program Files (x86)\1E\ActiveEfficiency\Scout\Scout.exe Modes=Server Servers=<PathToWindowsServerListFile>
```

Where `<PathToWindowsServerListFile>` should be replaced in the example to point to where your file containing a list of Windows servers resides, including the file name.

The file should contain a list of FQDN server host names, one per line in the file. For example:

```
ACMENTWK\DEV120-WACMEACE
DEV121-WACMEACE.AcmeNtwk.local
DEV122-WACMEACE.AcmeNtwk.local
ACMENTWK\DEV1234-WMKAL
DEV5678-WMKAL.AcmeNtwk.local
```

Scanning Linux servers

To scan Linux servers you use the `NixServers` command-line parameter and run a command similar to the following:

```
C:\Program Files (x86)\1E\ActiveEfficiency\Scout\Scout.exe Modes=Server NixServers=<PathToLinuxServerListFile>
```

Where `<PathToLinuxServerListFile>` should be replaced in the example to point to where your file containing a list of Linux servers resides, including the file name.

The file should contain a list of FQDN server host names, one per line in the file. For example:

```
ACMENTWK\DEV120-LACMEACE
DEV121-LACMEACE.AcmeNtwk.local
DEV122-LACMEACE.AcmeNtwk.local
ACMENTWK\DEV1234-LMKAL
DEV5678-LMKAL.AcmeNtwk.local
```

For Linux servers the Scout also needs credentials for an account that will be used to log onto each Linux server. These must be added to the Scout credentials file:

```
C:\Program Files (x86)\1E\ActiveEfficiency\Scout\Config\Credentials.config
```

The credentials file is in XML format. You set a credential pattern and password by addding an `<sshCredential>` tag to the `<sshCredentials>` tag. Each <sshCredential> tag must contain the following parameters:

Parameter	Description
matchingDevices	Set to a regular expression pattern to be used against the server names listed in the Linux server list file mentioned earlier. For example setting `matchingDevices="*"` will match against all the servers listed in the Linux server list file and will mean that the associated `username` and `password` will be tried for all listed servers.

username	Sets the username for the account.
password	Sets the password for the account.

The following example `Credentials.config` sets the Scout to try the `root` account with the password `X0123987` for all listed Linux servers, if logon with those credentials fails then the root account with the password `X0129386` will be tried for any listed server whose FQDN begins with the letters `Dev`.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <sshCredentials>
        <sshCredential matchingDevices="*" username="root" password="X0123987" />
        <sshCredential matchingDevices="Dev*" username="root" password="X0129386" />
    </sshCredentials>
</configuration>
```

Scanning both Windows and Linux Servers in a single scout command

You can put both parameters on the same command-line to get the scout to scan both Windows and Linux servers in one go.

```
C:\Program Files (x86)\1E\ActiveEfficiency\Scout\Scout.exe Modes=Server Servers=<PathToWindowsServerListFile>
NixServers=<PathToLinuxServerListFile>
```

Where `<PathToWindowsServerListFile>` should be replaced in the example to point to where your file containing a list of Windows servers resides and `<PathToLinuxServerListFile>` should be replaced in the example to point to where your file containing a list of Linux servers resides. Both of these should also include the file name.

Getting the data to AppClarity

When ActiveEfficiency has been configured you will then need to configure the synchronization of the ActiveEfficiency Connector on the server where the AppClarity Service resides. The AppClarity 5.2 documentation provides information on synchronizing the ActiveEfficiency Connector - please refer to AppClarity 5.2 - Connecting to ActiveEfficiency.