Limiting Number of Users Logged On

One of the common issues many portal projects have is to limit the number of users who can be logged on to the portal / WebAS (e.g. for WebDynpro) at any one time. This is a way to do it.

This is originally taken from my blog here: https://weblogs.sdn.sap.com/pub/wlg/5420. However, having it in a WIKI means that people can add their own experiences. Please do.

One of the common issues many portal projects have is to limit the number of users who can be logged on to the portal / WebAS (e.g. for WebDynpro) at any one time. This is normally due to the fact that the hardware can cope with normal everyday load, but is unable to cope with extreme peaks, such as input of timesheets or viewing of payslips.

First of all I will explain how the Java Web Application Server works in a normal scenario. When a user first logs on, your network switch should identify that the user does not have an existing session and load balance them to a WAS java dispatcher. The dispatcher will then load balance the new user session to one of it’s server nodes. For future requests (within the same session), the network switch should identify that you have an existing session and pass it to the same dispatcher (known as stickiness). The dispatcher will in turn pass it to the same server node. So in theory once you have a session, you should be sent to the same server node.

A service running on each server node is the Security Provider Service. When you use the Visual Administrator tool, go into Server -> Services -> Security Provider. In here it is possible to see the login sessions for that server node.

It is possible to set properties of the Security Provider. To limit the number of users, we use two properties, MaxUsersCount and SessionExpirationPeriod.

MaxUsersCount sets the number of users per server node, not per system. So simply divide the total number of users by the number of server nodes you are using.

SessionExpirationPeriod sets how long a user session remains on the server node if the user does not log off correctly (i.e. they click on X and DSM does not work correctly). It is very important that this is set else you will get ‘user creep’, i.e. the number of users reported to be logged on in the security provider is more than the actual number, so the limit will be reached early. This parameter is set in seconds and as default (on 640) is around 28 hours. So this must be reduced to 12 hours or less.

Note: During testing I have found that if a user uses the log off functionality, it removes them from the list of sessions. If they do not use the log off functionality, it doesn’t. So that is why it is important to set the SessionExpirationPeriod parameter.

More information on these parameters can be found on help.sap.com

So What Happens When The Limit is Reached?

When the limit is reached it returns the portal / WAS logon page. Unfortunately, there is currently a bug where it does not display an error message (due to be fixed in a later SPS). So if you enter a correct user id and password, it will just return the logon page again. I would recommend changing the logon screen a bit to have some sort of message to users (there are lots of blogs / forum messages explaining how to do this). Users who have a current session will not be affected as they already have a session on the server node.

Obviously the fact that it is just returns the logon page (including when using SSO on to the portal when using SPNego) is not ideal, but the opposite is that the system crashes due to too many users.

Another recommendation is to have one instance separate from the network switch load balancing (e.g. the central instance). The reason for this is so that administrators can still log on to this when all other instances are busy.

It is not the perfect solution, but does the job. A neater solution could possibly be done by writing your own login module.