

MSDN Blogs > Durgaprasad Gorti's WebLog > VSTT Controller and Agent setup

VSTT Controller and Agent setup



dgorti 2 Oct 2007 11:18 PM 4

In my previous blog article, I pointed out that in a workgroup environment, Windows XP has the force guest policy on and it prevents agents and clients connecting to the controller.

In this post, I want to expand on this and talk a bit more about VSTT controller and agent setup. Particularly, I will talk about the user account requirements on the machines running controller, agent and the client.

Ed Glas already talked about the setup in general at <http://blogs.msdn.com/edglas/pages/load-agent-and-load-controller-installation-and-configuration-guide.aspx>, where he has a list of user account and password requirements on the machines running controller and agent. In this post, I want to drill down further into the reasons behind those user account requirements. I will do this by walking you through the installation process of a controller, agent and a client and troubleshooting the authentication issues.

To start with, I have 3 machines. VSTTClient, VSTTController and VSTTAgent1. I have Windows XP SP2 on all the 3 machines. They are all in a workgroup TESTRIG. I have intentionally made the administrator password **different** on all the 3 machines. The idea is not accidentally authenticate to another box as an Admin. Other than this, all the machines are clean and no other setup has been done.

Before we proceed to the installation, I want to go over a few things.

1. Workgroup authentication

In a Windows domain environment, there is a central authority to validate credentials. In a workgroup environment, there is no such central authority. Still, we should be able to have computers in a workgroup talk to each other and authenticate users. To enable this, local accounts have a special characteristic that allows the local security authority on the computer to authenticate a "principal" in a special way. If you have two computers and a principal "UserXYZ" on both machines the security identifiers are different for MACHINE1\UserXYZ and MACHINE2\UserXYZ and for all practical purposes they are two completely different "Principals". However if the passwords are the same for them on each of these computers, the local security authority treats them as the same principal.

So when MACHINE1\UserXYZ tries to authenticate to MACHINE2\UserXYZ, and if the passwords are the same, then on MACHINE2, the UserXYZ is authenticated successfully and is **treated as MACHINE2\UserXYZ**. Note the last sentence. The user MACHINE1\UserXYZ is authenticated as MACHINE2\UserXYZ if the passwords are the same.

2. Force Guest

As I mentioned in the previous article, in a default Windows XP workgroup environment, the force guest policy is on by default. You can quickly check this out by mapping a share from a one machine to the other. Here, I am trying to map a share from the VSTTController to VSTTAgent1 and I am forced to login as a Guest, and the User Name box is grayed out.









If the force guest policy is on, it does not matter if you have the same users and passwords, you can not authenticate to the machine as any other principal than the Guest. You can turn it off using Regedt32, HKLM\System\CurrentControlSet\Control\Lsa and editing the forceguest value to 0.

3. Security Audit



If you are having authentication issues and don't know what is happening, the security audit policy is your good friend. Normally security audit is turned off and hence you will not see the events in the event viewer. To turn the security auditing, open mmc, add group policy object editor, select local computer. Then Select Local Computer Policy\Computer Configuration\Windows Settings\Security settings\Local Policies\AuditPolicy and turn on Success and Failure auditing for the events of interest. For troubleshooting purposes, I turn on the audit for everything here. Granted this generates a lot of auditing but at least I know

Options

-  Blog Home
-  Email Blog Author
-  Share this
-  RSS for posts
-  Atom
-  RSS for comments

Search Blogs

Search MSDN with Bing

 Search this blog  Search all blogs

Tags

[Ruby On Rails](#) [IIS](#) [IIS 7](#) [RoR](#) [Windows](#)

Archive

[December 2011](#) (1)
[May 2010](#) (1)
[September 2009](#) (1)
[June 2009](#) (3)
[January 2009](#) (1)
[October 2007](#) (1)
[September 2007](#) (2)
[October 2005](#) (2)
[September 2005](#) (4)
[August 2005](#) (1)

what is going on and once I am done troubleshooting, I can turn the auditing off.
Once done, on a command prompt, run gpupdate /force to apply the policy.

Installing the Controller and Agent

Now that we have the basics covered, I started the installation process with the VSTTController box on which I intend to install the Controller. The first thing is to turn off the force guest policy since I already know that the agent and the client needs to connect and authenticate to the controller. Then I added a user CSUser. This is the user the controller service will run as. During the installation process, you need to pick a user under which the controller service runs under and I picked the CSUser.

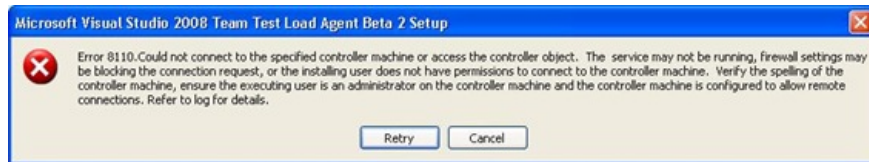
After the installation of the controller, I have, on the VSTTController machine,

- A user account CSUser
- An empty TeamTestAgentService group
- An empty TeamTestControllerAdmins group
- An empty TeamTestControllerUsers group.

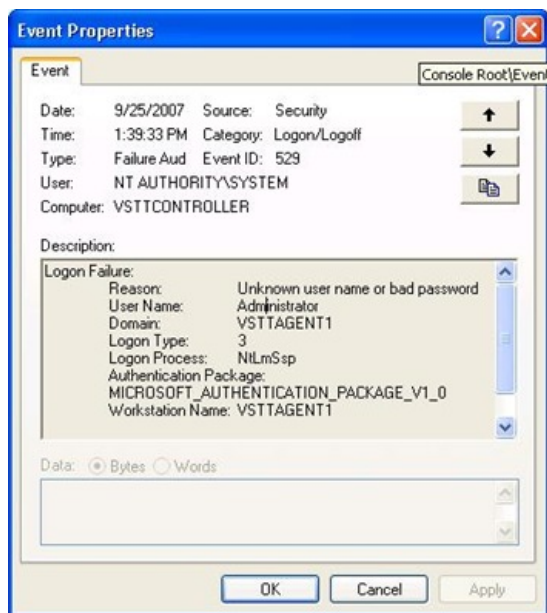
At this point the controller service seems to be running, so I am happy so far.

The next step then is to go to the VSTTAgent1 machine and install the Agent. Note that on this machine I did not yet turn the ForceGuest policy off on VSTTAgent1. The reason, once again, is that I wanted to understand the minimal stuff that I need to do get this rig working.

Before I started the agent installation, I added the user AgentUser to the machine. This is the identity under which the agent process will run. I logged in as **Administrator** on this box and started the agent installation. The installation process asks for the user identity of the agent process and I specified the identity as AgentUser. About 15 seconds into the process, the agent installation has trouble. Apparently connecting to the controller failed.



To start troubleshooting, I switched to the VSTTController machine, turn on the security auditing on the VSTTController machine as described above and then switched back to the VSTTAgent1 machine and clicked Retry on the dialog. As expected it fails again. Now when I switched to the VSTTController box and looked at the Event Viewer, under Security folder, I see the following event.



The VSTTController machine is rejecting the logon VSTTAgent1\Administrator. This is for good reason. The Administrator passwords are intentionally not the same!

It is easy to see why the agent installation connects to the controller box under the identity of the person who is installing the Agent. The user that is installing the agent is supposed to be part of the Administrators that administer the controller and agents. The act of installing an Agent and connecting to a controller makes the agent to be registered with the controller. This needs special permissions.

One thing we can do now is to adjust the administrator passwords to be the same. I don't particularly like the idea, since I don't want Administrators of one box to automatically connect to other boxes.

So what I decided to do is to add a user AgentAdmin on the VSTTAgent1 machine and add it to the Administrators group. The idea is that I will restart the Agent installation as AgentAdmin, not as Administrator. Since AgentAdmin is part of the Administrators group on VSTTAgent1 machine, the installation should work fine. On the VSTTController machine, I need to add the same user AgentAdmin with the same password. However VSTTController\AgentAdmin **will not be part of the Administrators group on VSTTController box**. The idea is to not unnecessarily grant permissions that are not required.

So at this point, here is what I have on each box

VSTTController	VSTTAgent1
ForceGuest Off	ForceGuest ON
AgentAdmin as a normal user	AgentAdmin user part of the Administrators group
CSUser as a normal user. The controller service is running under this user	AgentUser as a normal user. This is the intended identity under which the agent should be running
An empty TeamTestAgentService group	N/A
An empty TeamTestControllerAdmins group	
An empty TeamTestControllerUsers group	

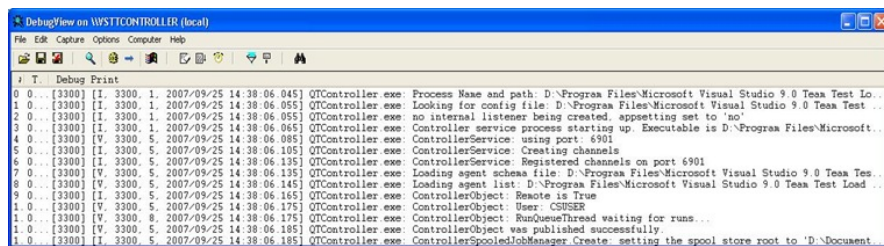
Security Audit turned on Security Audit off

With this arrangement, I logged off as an Administrator on the VSTTAgent1 box and then logged in back as a AgentAdmin.

But before I proceed, I will let you in on a cool trick. The .NET tracing's default trace listener writes the trace using the API called OutputDebugString and you can view the output in real time using a tool called DebugView. DebugView is a tool that is freely downloadable from the Technet. Go [here](#) to download. What this means is that you don't need to configure file based trace listeners, and keep refreshing the file editor to see the latest contents. Since I wanted to see the controller's trace messages, I switched to the VSTTController box, edited the QtController.exe.Config file (under <rootdrive>:\Program Files\Microsoft Visual Studio 9.0 Team Test Load Agent\LoadTest), and adjusted the EqTraceLevel to a value of "4" like shown below, saved the file and restarted the controller service for the settings to take effect.

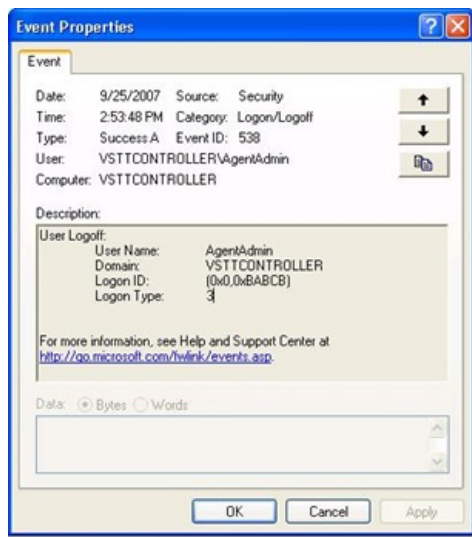
```
<switches>
<!-- You must use integral values for "value".
    Use 0 for off, 1 for error, 2 for warn, 3 for info, and 4 for verbose. -->
<add name="EqTraceLevel" value="4" />
</switches>
```

Once I did this, I fired up the DebugView. You should see something like this on the VSTTController box.



Now that I have both the security audit and the controller tracing on the VSTTController box, I came back to the VSTTAgent1 box and started the install process. Remember that now I am logged in on the VSTTAgent1 Box as AgentAdmin, which is part of the VSTTAgent1\Administrators group.

The Agent installation fails again. On the VSTTController box, I look at the event viewer. I see a successful logon of the AgentAdmin. The Logon type = 3 which means that this is a network logon. So we made progress. We are able to authenticate to the VSTTController machine as VSTTController\AgentAdmin from VSTTAgent1 machine using VSTTAgent1\AgentAdmin principal.



The next thing to look at is the QtController's trace messages from the DebugView tool. The trace messages show that the controller service is checking for group membership of this authenticated principal (user) AgentAdmin. It is checking to see if the AgentAdmin is a member of TestTestControllerUsers or TeamTestControllerAdmins group. Since I did not add the AgentAdmin as part of any group on the VSTTController box, the membership check fails and controller returns an error which is showing up as an installation failure on VSTTAgent1 machine.

To fix the error it is not clear to me as to what group I need to add the AgentAdmin user to on the VSTTController box. At first I tried adding the AgentAdmin to the TeamTestControllerUsers group, went back to the VSTTAgent1 box and retried the installation. Even though the installation was successful, the agent was never registered with the controller. I could have tried to fix through other means, but decided to try this through the user scenario. So I added the AgentAdmin as part of the TestTestControllerAdmins group on the VSTTController box. I turned the security auditing on on the VSTTAgent1 machine too, just to aid in debugging.

I uninstalled the agent from the VSTTAgent1 box and attempted a reinstall of the Agent. Here is what I have just before the reinstall of the Agent on VSTTAgent1 machine

VSTTController	VSTTAgent1
ForceGuest Off	ForceGuest ON
AgentAdmin as a normal user and part of the TeamTestControllerAdmins	AgentAdmin user part of the Administrators group
CSUser as a normal user. The controller service is running under this user	AgentUser as a normal user. This is the intended identity under which the agent should be running
Empty TeamTestAgentService group	N/A
TeamTestControllerAdmins group has AgentAdmin as a member	
Empty TeamTestControllerUsers	

Security Audit turned on	Security Audit ON
--------------------------	--------------------------

The installation seems to be successful. On the VSTTController machine I see trace messages showing that the installation went well and the VSTTAGENT1 is added to the AgentManager. So far so good.

Now that the installation of the agent is successful, I really don't see the point of having the AgentAdmin as a user on the VSTTController machine or on the VSTTAgent1 machine. I want to delete this account so that I can reduce any unnecessary user accounts with the same passwords floating around. So I deleted the AgentAdmin account from both the VSTTController and VSTTAgent1 machines (first I need to logoff as AgentAdmin on the VSTTAgent1 box, log back in as Administrator before deleting the AgentAdmin user). With the AgentAdmin account deleted from both the machines, here is the configuration

VSTTController	VSTTAgent1
ForceGuest Off	ForceGuest ON
CSUser as a normal user. The controller service is running under this user	AgentUser as a normal user. This is the intended identity under which the agent should be running

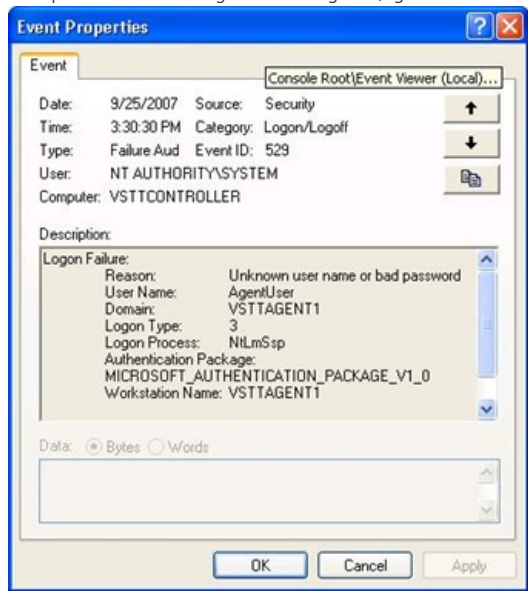
Empty TeamTestAgentService group N/A
Empty TeamTestControllerAdmins
group
Empty TeamTestControllerUsers group

Security Audit turned on Security Audit **ON**

The Agent installation suggested a reboot after the installation so I did that.

After the reboot when I logged into the VSTTAgent1 box as an Administrator and opened the event viewer. I am expecting that the Agent service will not be able to connect to the controller. Why? The agent is running as VSTTAgent1\AgentUser and there is no corresponding user on the VSTTController box. So Agent should fail to connect.

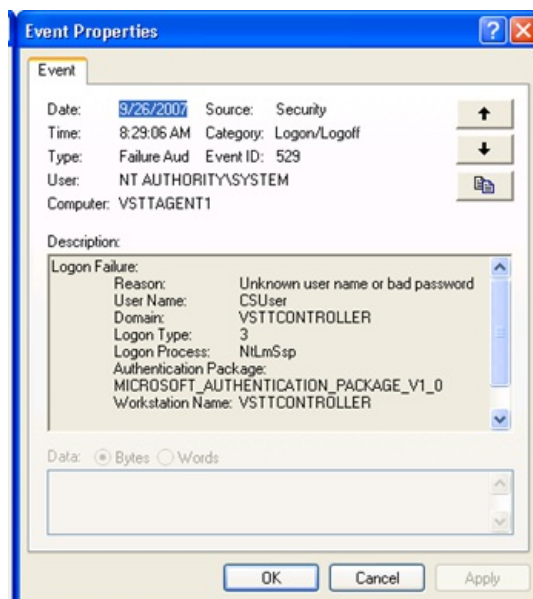
Sure enough the event viewer on VSTTAgent1 machine and the event viewer on the VSTTController machine both point to the failed logon of VSTTAgent1\AgentUser on VSTTController machine.



Since we need the agent to connect to controller, we must add the AgentUser to the VSTTController machine with the same password as it has on VSTTAgent1 machine.

Even after adding this, the agent is still not happy since it still can't connect to the controller service. Looking at the debug view, you can figure out that the AgentUser needs to be part of the TeamTestAgentService group. So I added the AgentUser to the TeamTestAgentService group. Still no success.

Looking at the Event viewer on VSTTAgent1, I find that the VSTTController\CSUser is trying to authenticate to VSTTAgent1 machine. This is because the controller wants to callback to the agent on a different connection and that connection needs to be authenticated.



The auth fails due to the fact that we have force guest on and we don't have CSUser as a local user on VSTTAgent1. Let's fix this now. After fixing, and rebooting, this is what I have for configuration

VSTTController	VSTTAgent1
ForceGuest Off	ForceGuest Off
CSUser as a normal user. The controller service is running under this user	CSUser as a normal user.

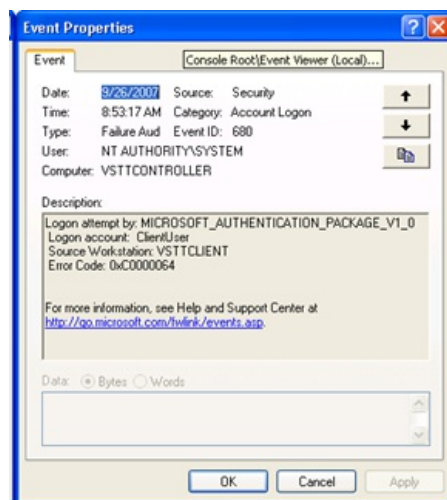
The AgentUser as a normal user. AgentUser as a normal user. This is the intended identity under which the agent should be running

TeamTestAgentService group contains N/A
the AgentUser
Empty TeamTestControllerAdmins
TeamTestControllerUsers group contains AgentUser

Security Audit turned on Security Audit **ON**
That does it. The agent and controller are now happy and are communicating to each other.

Installing the Client and getting it to connect to the controller

I created a ClientUser on the VSTTClient machine. I installed VSTT, logged off and logged in back again a VSTTClient\ClientUser. I fired up VS, created a test project, specified VSTTController as the remote controller and started a run. As expected, the VS is trying to open an authenticated connection to the VSTTController as VSTTClient\ClientUser. Since that user is not defined on the VSTTController machine, the login fails.



The fix is to add the ClientUser to the VSTTController box and then to the TeamTestControllerUsers group. Even after doing that you would still find that you are able to execute tests. Following the same troubleshooting techniques and turning on the security audit on the VSTTClient, the VSTTController\CSUser is failing to authenticate to the VSTTClient. So the fix is to turn off the force guest policy and create the CSUser on the VSTTClient machine.

Now I can run tests in my multi machine workgroup topology....

Here is the final recipe for the VSTT Client, Controller and agent setup

1. On all machines in the workgroup, turn off the ForceGuest policy.
2. Determine the user account under which the controller is going to run under [CSUser]
Add this account to *all* the machines - client, controller and agent machines [VSTTClient, VSTTController, and VSTTAgent1 in my case]
3. On the controller machine, install the controller. When you install the controller it will create a set of user groups mentioned above.
4. Determine the account under which the agent is going to run under [AgentUser]. Add the account to the controller machine and to the TeamSystemAgentService Group.
5. On the agent machine, create a temporary account called AgentAdmin. Add the account to the Administrators group. Add the AgentAdmin account to the controller machine and to the TeamSystemControllerAdmins user group
6. Login to the Agent machine as the AgentAdmin
Install the agent, specify AgentUser as the user under which the server is going to run under
7. Once the install is complete, on the Agent machine, logoff as AgentAdmin, log back in as Administrator and delete the AgentAdmin account. Delete the AgentAdmin account on the Controller machine. There is no need for this account any more.
8. Determine the account under which you are going to run the Visual Studio on the client machine. [ClientUser on VSTTClient]. Add the account to the controller machine with the same password. On the controller machine, add the client user to the TeamSystemControllerUsers group.

Thats it. I hope this article showed you in detail how to install controller and agent in a workgroup and troubleshoot any authentication issues using the DebugView and Security Audit policy.



Comments





Florian Engeser 2 Oct 2007 11:35 PM #
PingBack from <http://www.artofbam.com/wordpress/?p=4748>



Florian Engeser 3 Oct 2007 10:08 AM #
Brian Harry on Updates to TFS 2008 Beta 2 Brian Krieger on Orcas Beta 2 Refresh is now available. Jeff...



Florian Engeser 4 Oct 2007 10:15 AM #
ASP.Net, C#, .Net Framework, SQL Server: Error Handling in .Net with Example Great article on error...



Florian Engeser 5 Oct 2007 1:41 PM #
Check out Durga's blog post on setting up an agent in workgroup mode . One thing I noticed is that the

Leave a Comment

Name

Comment

Post