Show

Configuring DCOM for Remote Access

Before you can run any J-Integra® application in DCOM mode (i.e. the Java application and the COM application are located on two separate machines), you must ensure that DCOM is properly configured. Failure to do so typically results in one of the following errors:

- *AutomationException: 0x80070005 - General access denied error*
- *AutomationException: 0x5 - access is denied*
- *Run-time error '70': Permission denied*

**** Configuring DCOM is mandatory when running J-Integra® in DCOM mode. However, if you are using native mode, DCOM configuration is seldom necessary (since standard COM is used for communication between components, not DCOM). However, if you are running in native mode and you are still getting "access denied" errors, please configure DCOM as instructed below.***

Contents

DCOM Authentication

Before you can access a COM component via DCOM, you must provide the authentication credentials of a user who has been granted permission to launch/access the component. J-Integra® allows you to do this using native code or by specifying the user credentials in your Java code.

1. Native Authentication

By default, J-Integra® will use native code to determine the user credentials of the user who is currently logged in. This will only work if the following conditions are met:

- The Java client resides on a Windows machine. If the Java client is located on a non-Windows machine, you must use *AuthInfo.setDefault()* to provide authentication credentials (see below).
- The ***%JINTEGRA_HOME%\bin*** directory must be in the system PATH, and ***ntvauth.dll*** must reside there. This DLL contains the native code J-Integra® uses to determine the local authentication credentials.
- If the Java client is running on machine A, then the credentials of the user who is currently logged on to machine A must be configured on machine B to have access/launch permissions to the COM component being accessed. For more information on DCOM Configuration, see below.

2. AuthInfo.setDefault

If you are running on a non-Windows platform, or you wish to use the credentials of someone other than the user who is currently logged in, then you need to use the method ***AuthInfo.setDefault()***. This identitiy will be used on a process-wide basis when accessing and using COM components.To use this method, place the following line in your Java client before any calls to the COM component:

```
com.linar.jintegra.AuthInfo.setDefault("DOMAIN", "USER", "PASSWORD");
```

* Using *AuthInfo.setDefault()* will only work if the Windows machine hosting the COM component belongs to a domain or is a domain controller.

DOMAIN - The domain to which the user belongs. If the user does not belong to a domain, then pass the name of the machine to which the user belongs. Note that *COMPUTER\USER* and *DOMAIN\USER* are two completely different security principles. For more information, please refer to Microsoft's COM Security FAQ.
USER - The username being used to access the COM component.
PASSWORD - The password of the above username.

You may override the process-wide default using ***AuthInfo.setThreadDefault()***. This method establishes the authentication to be used for the **current thread** only. To clear the authentication for the current thread, call ***AuthInfo.setThreadDefault(null)***. Please note that ***setThreadDefault()*** will behave differently if the COM component is a *"single instance"* component. For more information, see AuthInfo.setThreadDefault(...) and Single-Instance COM Objects in the Knowledge Base.

Regardless of whether or not you set the authentication on a per-thread basis, it is strongly recommended that you call ***AuthInfo.setDefault()*** early in your Java code to establish the authentication to be used on a process-wide basis. This is needed to allow J-Integra® daemon threads to perform authenticated communications (such as when releasing COM object references that have been garbage collected).

3. DCOM Authentication for Usernames with Non-ANSI Characters

Because Java supports non-ANSI characters, you can use domain/username/password combinations with non-English characters. In order to use non-English characters, you need to make sure that the same non-English language is set as the default language on both the computer running the Java client and the computer hosting the COM component.

Follow the steps below to configure the Windows 2000/XP English version machine hosting the COM component:

1. Click **Start**, click **Control Panel**, click **Regional and Language Options**.
2. Click **Advanced**, select your language in **Code page conversion tables**, and select your language from the **Language for non-Unicode programs** dropdown combo box. This will set the new default language and allow non-English characters.
3. Click **OK** button. Then reboot the machine for the settings to take effect.
4. Configure DCOM to grant DCOM access/launch permissions to the non-English username.

Follow the steps below to configure the Windows NT English version machine hosting the COM component:

1. Click **Start**, click **Control Panel**, click **Regional Settings**.
2. Click **Reginal Settings** tab, select your language in from the **Many programs support international settings...** dropdown combo box, check **Set as system default local** checkbox. Click **Input Locals** tab, add your language to **Installed input locals and layouts** list box, select the language you just added, and click **Set as Default** button.
3. Click the **OK** button. Then reboot the machine for the settings to take effect.
4. Configure DCOM to grant DCOM access/launch permissions to the non-English username.

If your Java client machine is also Windows, follow steps 1-3 above to configure the machine. You can then pass non-English characters to ***com.linar.jintegra.AuthInfo.setDefault()***.

Please refer to your platform manual to set the default language if your Java client machine is a non-Windows platform.

DCOM Configuration (DCOMCNFG)

DCOMCNFG is a tool that comes with Windows that allows users to configure the DCOM settings of a COM application. J-Integra® provides its own DCOM configuration tool called DCOMConfig.exe. J-Integra®'s DCOMConfig tool is a simplified version of Window's DCOMCNFG meant for users who are new to J-Integra® and/or DCOM. If you'd rather use DCOMConfig.exe, please refer to the tool description.

We recommend the following DCOM configuration settings as a starting point if you are unfamiliar with J-Integra® and/or DCOM. Once you are more comfortable with DCOM configuration, feel free to modify your settings based on your own set of requirements.

Configuring DCOM on Windows 2000

1. Click **Start**, click **Run**, and then type **DCOMCNFG**.
2. Click **Default Properties**. Select **Enable Distributed COM on this computer**. Set the **Default Authentication Level** to **Connect** (**None** also works). Set the **Default Impersonation Level** to **Identify** (**Impersonate** also works).

   Here are the rules regarding **Authentication Level** for J-Integra® applications:

   - The Authentication Level doesn't matter if you are running in native mode.
   - The machine hosting the COM client/server application must be set to ***Connect*** or ***None*** level authentication, as this is all that J-Integra® supports. It doesn't matter what the Authentication Level is on the machine running J-Integra®.
   - For COM-accessing-Java applications, the Authentication Level on the COM client machine must be set to ***Connect*** or ***None***.
   - For Java-accessing-COM applications, the Authentication Level on the COM server machine must be set to ***Connect*** or ***None***.
3. Click **Default Security**.
4. Under **Default Access Permissions** click **Edit Default**. Add **SYSTEM** and **INTERACTIVE**. The user whose authentication credentials will be used to access the COM application must also be included in this list. There are many ways to do this. You can add the specific user or simply add a group the user belongs to. Possible values include:

   - *Domain\Username* (A specific user)
   - *Domain\Administrators* (All administrators on a specific domain)
   - *Everyone* (All users)
5. Under **Default Launch Permissions** click **Edit Default**. Make sure the **Default Launch Permissions** have the same values as the **Default Access Permissions**.
6. Click **Default Protocols**. Make sure **Connection-oriented TCP/IP** is listed first.
7. You must now configure the COM application you wish to access. Click **Applications** and right-click on the application you wish to configure. Select **Properties**. If your COM application is a DLL, you must first create a surrogate EXE for it using the SetDllHost tool. Once a surrogate EXE is created, the surrogate name will appear in the list of applications. Select **Properties** for the surrogate and continue on.
8. Click **General**. Set the **Authentication Level** to **Default**.
9. Click **Location**. Select **Run application on this computer**.
10. Click **Security**. Select **Use default access permissions** and **Use default launch permissions**.
11. Click **Identity**. Select **The launching user**. This setting specifies the account that will be used to run the COM application once it is launched by a client program. **The launching user** is the user account of the client process that launched the server, and is the recommended setting. Depending on the COM application you want to connect to, you may need to change this to:
    - **The interactive user** - The user that is currently logged on to the machine hosting the COM application (use this if you are going to access MS Excel and make it visible).
    - **This user** - Specify a user account that will always be used to run the COM application regardless of which user is accessing it.

    For more information on *"How To Configure Office Applications to Run Under the Interactive User Account"* (which includes information on using Terminal Services), please see the References section at the bottom of this page.
12. Click **Endpoints**. Select **default system protocols**.
13. If you still get an "access denied" or "permission denied" error after configuring your DCOM settings, try rebooting your machine to allow the new settings to take effect.

Configuring DCOM on Windows XP and Windows Server 2003

1. If the computer belongs to a workgroup instead of a domain, make sure that it does not use simple file sharing. Open **Windows Explorer** or double click **My Computer**, click **Tools**, then go to **Folder Options**, click **View** and uncheck **Use simple file sharing (Recommended)** in **Advanced settings**.

2. Click **Start**, click **Programs**, click **Administrative Tools**, click **Component Services**.

3. Expand **Component Services**, expand **Computers**, and right-click **My Computer**. Select **Properties**.

4. Click **Default Properties**. Select **Enable Distributed COM on this computer**. Set the **Default Authentication Level** to **Connect** (**None** also works). Set the **Default Impersonation Level** to **Identify** (**Impersonate** also works).

   Here are the rules regarding **Authentication Level** for J-Integra® applications:

   - The Authentication Level doesn't matter if you are running in native mode.
   - The machine hosting the COM client/server application must be set to ***Connect*** or ***None*** level authentication, as this is all that J-Integra® supports. It doesn't matter what the Authentication Level is on the machine running J-Integra®.
   - For COM-accessing-Java applications, the Authentication Level on the COM client machine must be set to ***Connect*** or ***None***.
   - For Java-accessing-COM applications, the Authentication Level on the COM server machine must be set to ***Connect*** or ***None***.

5. Click **Default COM Security**.

6. Under **Default Access Permissions** click **Edit Default**. Add **SYSTEM**, **INTERACTIVE**, and **NETWORK**. The user whose authentication credentials will be used to access the COM application must also be included in this list. There are many ways to do this. You can add the specific user or simply add a group the user belongs to. Possible values include:

   - *Domain\Username* (A specific user)
   - *Domain\Administrators* (All administrators on a specific domain)
   - *Everyone* (All users)

7. Under **Default Launch Permissions** click **Edit Default**. Make sure the **Default Launch Permissions** have the same values as the **Default Access Permissions**.

8. Click **Default Protocols**. Make sure **Connection-oriented TCP/IP** is listed first.

9. You must now configure the COM application you wish to access. Expand **Component Services**, expand **Computers**, expand **My Computer**, and click **DCOM Config**. Right-click on the application you wish to configure. Select **Properties**. If your COM application is a DLL, you must first create a surrogate EXE for it using the SetDllHost tool. Once a surrogate EXE is created, the surrogate name will appear in the list of applications. Select **Properties** for the surrogate and continue on.

10. Click **General**. Set the **Authentication Level** to **Default**.

11. Click **Location**. Select **Run application on this computer**.

12. Click **Security**. Set **Launch Permissions** to **Use Default**. Set **Access Permissions** to **Use Default**. Set **Configuration Permissions** to **Use Default**.

13. Click **Identity**. Select **The launching user**. This setting specifies the account that will be used to run the COM application once it is launched by a client program. **The launching user** is the user account of the client process that launched the server, and is the recommended setting. Depending on the COM application you want to connect to, you may need to change this to:
    - **The interactive user** - The user that is currently logged on to the machine hosting the COM application (use this if you are going to access MS Excel and make it visible).
    - **This user** - Specify a user account that will always be used to run the COM application regardless of which user is accessing it.

    For more information on *"How To Configure Office Applications to Run Under the Interactive User Account"* (which includes information on using Terminal Services), please see the References section at the bottom of this page.

14. Click **Endpoints**. Select **default system protocols**.

15. If you still get an "access denied" or "permission denied" error after configuring your DCOM settings, try rebooting your machine to allow the new settings to take effect.

Configuring DCOM on Windows XP SP2

Microsoft has added some DCOM security enhancements to Windows XP Service Pack 2. In addition to the above Windows XP DCOM configuration settings, you will need to perform the following steps.

1. If the computer belongs to a workgroup instead of a domain, make sure that it does not use simple file sharing. Open **Windows Explorer** or double click **My Computer**, click **Tools**, then go to **Folder Options**, click **View** and uncheck **Use simple file sharing (Recommended)** in **Advanced settings**.
2. Click **Start**, click **Programs**, click **Administrative Tools**, click **Component Services**.
3. Expand **Component Services**, expand **Computers**, and right-click **My Computer**. Select **Properties**.
4. Click **Default COM Security**.
5. Under **Default Access Permissions** click **Edit Default**. Make sure **SYSTEM**, **INTERACTIVE**, **NETWORK**, and the user whose authentication credentials will be used to access the COM application all have **Local and Remote Access** permissions.
6. Under **Default Access Permissions** click **Edit Limits**. Service Pack 2 comes with the following default values: **ANONYMOUS LOGON** (Local Access) and **Everyone** (Local and Remote Access). Make sure these values are listed, and then add the user whose authentication credentials will be used to access the COM application. Allow this user to have **Local and Remote Access** permissions.
7. Under **Default Launch Permissions** click **Edit Default**. Make sure **SYSTEM**, **INTERACTIVE**, **NETWORK**, and the user whose authentication credentials will be used to access the COM application all have **Local and Remote Launch** permissions, as well as **Local and Remote Activation** permissions.
8. Under **Default Launch Permissions** click **Edit Limits**. Service Pack 2 comes with the following default values: **MACHINE\Administrators** (Local and Remote Launch, Local and Remote Activation) and **Everyone** (Local Launch and Local Activation). Make sure these values are listed, and then add the user whose authentication credentials will be used to access the COM application. Allow this user to have **Local and Remote Launch** permissions, as well as **Local and Remote Activation** permissions.
9. Service Pack 2 comes with a built-in Windows Firewall. If the firewall is turned on, you will have to allow your COM application network access to your machine. You can do this by opening Windows Firewall and adding your COM application to the list of programs under the **Exceptions** tab. If **Display a notification when Windows Firewall blocks a program** is selected, then you will be prompted to unblock the COM application when you run your J-Integra® application the first time. Select **Unblock** when prompted.
10. If you still get an "access denied" or "permission denied" error after configuring your DCOM settings, try rebooting your machine to allow the new settings to take effect.

For more information on DCOM security enhancements in Microsoft Windows XP Service Pack 2, see here.

Configuring DCOM on Windows 98

DCOM does not come pre-installed with Windows 98. If you wish to use J-Integra® on Windows 98, you must download DCOM98 from Microsoft. In addition, Windows 98 does not support the automatic launch of a server. Therefore, you must manually start the COM server you wish to access before running your Java client. If you do not do this, you will get the following error message:

> ***AutomationException: 0x80080005 - Server execution failed. Note that Windows 95 does not support automatic launch of a server, it must be running already.***

Network Security: LAN Manager Authentication Level

Microsoft Windows has a setting that determines which challenge/response authentication protocol is used for network logons. This choice affects the level of authentication protocol used by clients, the level of session security negotiated, and

the level of authentication accepted by servers.

J-Integra® only supports ***Send LM & NTLM responses***. Since this is the default setting for Windows platforms, this is typically not a problem for most users. However, if your network administrator has changed this setting you will probably get an error similar to:

> *java.io.IOException: Unable to establish RPC Connection to DCOM SCM on 192.168.4.202 (Bind returned Bind_NAK)*

Please ensure that the LAN Manager Authentication Level is set to ***Send LM & NTLM responses*** by doing the following:

1. Click **Start**, click **Settings**, click **Control Panel**, click **Administrative Tools**, click **Local Security Policy**.
2. In the **Local Security Settings** dialog box, click **Local Policies**, and then click **Security Options**.
3. Double-click **Network Security: LAN Manager Authentication Level**.
4. Select **Send LM & NTLM responses**.
5. Click **OK** and exit out of **Local Security Settings**.

Since J-Integra® implements the NT Challenge-Response Mechanism (NTLM) in pure Java as part of its DCOM engine, user passwords are **NEVER** sent over the wire. Microsoft's own documentation verifies this:

> *Is my password being sent across the network during NTLM authentication?*
> *No. NTLM authentication does not send the user's password (or the hashed representation of the password) across the network. Instead, NTLM authentication uses a challenge/response mechanism to ensure that the actual password never traverses the network.*
> *http://www.microsoft.com/technet/security/bulletin/MS01-001.mspx*

* J-Integra® only supports ***Authentication***, not ***Encryption***.

Accessing COM Components Within ATL Services

By default, the Microsoft ATL Service Wizard generates code which initialises all COM components running within the service to use ***Packet*** level authentication. J-Integra® only supports ***Connect*** level authentication (or ***None***), so you will need to modify the generated code as follows.

The following generated code should be changed from this...

> *hr = CoInitializeSecurity(sd, -1, NULL, NULL, **RPC_C_AUTHN_LEVEL_PKT**, RPC_C_IMP_LEVEL_IMPERSONATE, NULL, EOAC_NONE, NULL);*

To the following...

> *hr = CoInitializeSecurity(sd, -1, NULL, NULL, **RPC_C_AUTHN_LEVEL_CONNECT**, RPC_C_IMP_LEVEL_IMPERSONATE, NULL, EOAC_NONE, NULL);*

References

If you are using Windows Terminal Services, please read...

> **How To Configure Office Applications to Run Under the Interactive User Account**
> *http://support.microsoft.com/default.aspx?scid=kb;EN-US;288366*

If you are configuring Microsoft Office applications server-side (e.g. JSP/Servlet accessing Microsoft Excel), please read...

> **INFO: Considerations for Server-Side Automation of Office**
> *http://support.microsoft.com/default.aspx?scid=kb;EN-US;257757*