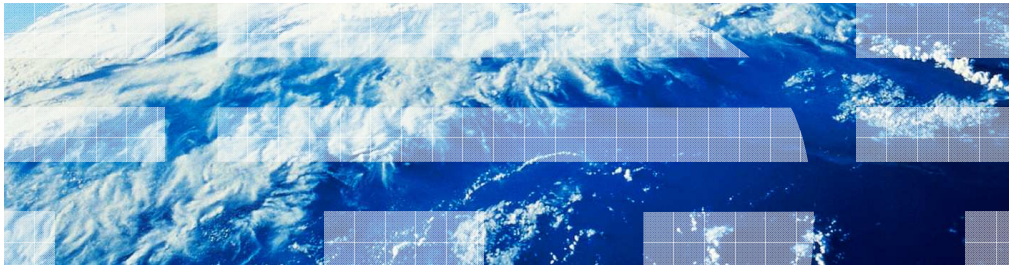


Configuring DataStage PX to use ssh instead of rsh



This presentation describes the steps needed to configure DataStage® PX to use ssh instead of rsh. This presentation assumes that ssh servers have been installed on all the machines where PX is to run.

Objectives

- System requirements
- Generating public and private RSA or DSA key pair
- Sending public key to remote node
- Adding public key to list of authorized keys
- Creating remsh file

The objective of this presentation is to describe how to generate the required public and private RSA or DSA key pair and how to send the generated keys to the remote node. You will also learn how to add the generated keys to the list of authorized keys, and how to create the remsh file.

System requirements

- PX already installed on remote nodes
- Configure ssh to run on conductor node
- Uses RSA or DSA public key encryption
- Tasks must be completed for each DataStage PX user

This presentation assumes that you have already successfully installed the PX engine on all of your remote nodes. In order for DataStage PX to use ssh, ssh must be configured so that a command can be launched from the conductor node to all other nodes without a password; authenticating only by way of public key encryption. When you configure ssh to work with the parallel engine, the engine connects from the primary computer to all of the other computers, using RSA or DSA public key encryption for authentication. This task must be completed for each user that runs parallel jobs. In the next steps, the primary computer, or the Conductor Node, is the computer that contains the IBM Information Server engine. The secondary computers, or Remote Nodes, are the other computers that contain parallel engines.

RSA versus DSA pairs

- RSA or DSA pairs generated depending on protocol version ssh is supporting
- RSA – ssh set to use Protocol 1
- DSA – ssh set to use Protocol 2
- Default settings
 - /etc/ssh/ssh_config
 - \$HOME/.ssh/config
- Protocol 2,1

The next step is to generate an RSA or DSA key pair. Which one you will generate depends on the protocol version ssh is using on your nodes. If you are using protocol 1 then you will want to generate an RSA key pair. If you are using protocol 2, then generate a DSA key pair. Your system will have the default settings in /etc/ssh/ssh_config. The user can also have this set in \$HOME/.ssh/config. Look in these files for the line with “Protocol” to see what the default setting is. If the default is “2,1”, for example, ssh will try version 2 and falls back to version 1 if version 2 is not available.

If you are unsure which protocol you are using, you can generate both the RSA and DSA keys for each remote machine.

Generating RSA key pair

- Generate public and private RSA key pair on conductor
 - `ssh-keygen -b 1024 -t rsa -f $HOME/.ssh/id_rsa`

Command Output:

```
Generating public/private rsa key pair.
Enter file in which to save the key ($HOME/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in $HOME/.ssh/id_rsa.
Your public key has been saved in $HOME/.ssh/id_rsa.pub.
The key fingerprint is:
f6:61:a8:27:35:cf:4c:6d:13:22:70:cf:4c:c8:a0:23 dsadm@conductornode
```

- Send public key to each remote node
 - `scp id_rsa.pub user@remotenode:$HOME/.ssh`

Login to the conductor node as the user that you will be running DataStage PX jobs as. Generate the public and private RSA key pair on the conductor node using the `ssh-keygen` command where `$HOME` is your user's home directory: `ssh-keygen -b 1024 -t rsa -f $HOME/.ssh/id_rsa`.

Press enter twice to set a null pass-phrase. The identification keys will have been saved in `$HOME/.ssh/id_rsa`. Next, ensure that you are in `$HOME/.ssh`. Send the public key to each of the remote nodes using the `scp` command, `scp id_rsa.pub user@remotenode:$HOME/.ssh`

where `remotenode` is the name of your remote PX node. You must do this step for every remote node that you are configuring.

Generating DSA key pair

- Generate public and private DSA key pair on conductor
 - `ssh-keygen -b 1024 -t dsa -f $HOME/.ssh/id_dsa`

Command Output:

Generating public/private dsa key pair.
Enter file in which to save the key (\$HOME/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in \$HOME/.ssh/id_dsa.
Your public key has been saved in \$HOME/.ssh/id_dsa.pub.
The key fingerprint is:
f6:61:a8:27:35:cf:4c:6d:13:22:70:cf:4c:c8:a0:23 dsadm@conductornode
- Send public key to remote node
 - `scp id_dsa.pub user@remotenode:$HOME/.ssh`

Login to the conductor node as the user that you will be running DataStage PX jobs as. Generate the public and private DSA key pair on the conductor node using the `ssh-keygen` command where `$HOME` is your user's home directory:

```
ssh-keygen -b 1024 -t dsa -f $HOME/.ssh/id_dsa.
```

Press enter twice to set a null pass-phrase. The identification keys will have been saved in `$HOME/.ssh/id_dsa`. Next, ensure that you are in `$HOME/.ssh`. Send the public key to the remote node using the `scp` command where `remotenode` is the name of your remote PX node. You must do this step for every remote node that you are configuring.

Adding key pair to authorized keys

- Logon to remote nodes
 - sh remotenode
- Add public key to list of authorized keys
 - Execute on remote node
 - cd \$HOME/.ssh
 - cat id_rsa.pub id_dsa.pub >> authorized_keys2
 - chmod 640 authorized_keys2
 - rm -f id_rsa.pub id_dsa.pub
- Test connection (From conductor node)
 - ssh dbnode ls

Next, log in to the remote node and add the public key to the list of authorized keys. Change directories into the .ssh directory under the user's home directory. In there you should find the key pair you created in the last step. You want to append the key pair to the authorized_keys file. If you created both the RSA and DSA key pairs, you want to append both to the file. In the example displayed on this slide, you used both the RSA and DSA key pairs. Be aware the file containing the authorized keys file can either be named authorized_keys or authorized_keys2, depending on the version of ssh you have installed on your systems.

Test the connection by trying to ssh from the conductor node to the remote node. If successful, you should be able to log in to the remote shell without providing a password. An example command to use as a test is ssh dbnode ls where dbnode is the name of the remote node you are testing.

Creating remsh file on conductor node

- Navigate to
IBM/InformationServer/Server/PXEngine/etc
- Create file called remsh which contains
 - `#!/bin/sh`
 - `exec /usr/bin/ssh "$@"`
- Give file execute permissions
 - `chmod 755 remsh`

Next, on the conductor node, navigate to the IBM/InformationServer/Server/PXEngine/etc directory. Create a file called remsh which contains:

```
#!/bin/sh
```

```
exec /usr/bin/ssh "$@"
```

An example remsh file which uses rsh can be found at PXEngine/etc/remsh.example. Give the file execute permissions by running: `chmod 755 remsh`.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

[mailto:iea@us.ibm.com?subject=Feedback_about_Relocating the Xmeta Respository.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_Relocating_the_Xmeta_Respository.ppt)

This module is also available in PDF format at: [../Relocating the Xmeta Respository.pdf](#)

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and DataStage are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.