



Agentless scheduling with Tivoli Workload Scheduler

Tivoli. software



© 2009 IBM Corporation
Updated July 21, 2009

In this session you will learn what agentless scheduling means, how to install agentless scheduling, and how to use it in your workload environment.

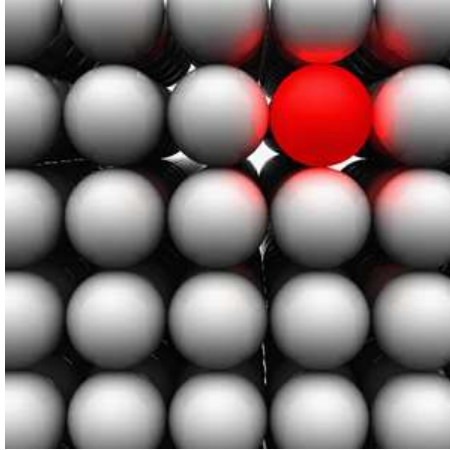
Agenda

- What is Agentless
- Agent versus Host
- User mapping
- Passwordless logins
- Workload Scheduler configurations
- Scheduling and executing
- Customizations



During the session you will learn what agentless scheduling means and the components of Tivoli Workload Scheduler that are involved. You will also learn about mapping user IDs, configuring Workload Scheduler, scheduling and executing jobs, and customizing the agentless scheduling.

What is Agentless



- Provided with Tivoli Workload Scheduler 8.4 or later
- Unixssh method
- SSH
 - ▶ Secure shell
 - ▶ UNIX® and Linux®

Starting with IBM Tivoli Workload Scheduler 8.4, a new type of agent was introduced, the unixssh method. Unixssh is a method for running work on computers that do not have Tivoli Workload Scheduler agents installed, thus, the term agentless scheduling. Unixssh uses Secure Shell to run jobs remotely.

Secure Shell or SSH is a network protocol for data exchange using a secure channel between two devices that are networked. Used primarily on Linux-based and Unix-based computers to access shell accounts, SSH was designed as a replacement for TELNET and other unsecured remote shells. These unsecured shells send information, notably passwords in plain text, leaving them open for interception. The encryption used by SSH provides confidentiality and integrity of data over an unsecured network such as the Internet.

Advantages

- No Tivoli Workload Scheduler (TWS) installation and maintenance
- Very lightweight software footprint
- No constant network connection or linking
- Very low bandwidth

Using agentless technology to run work remotely with Workload Scheduler has many advantages.

Maintenance of the software is minimal, if non-existent, because there is no agent to install.

The Tivoli Workload Scheduler agentless method creates only a small wrapper script on the agentless computer, minimizing the disc space requirement.

Agentless computers are not linked to the Tivoli Workload Scheduler network the way that Workload Scheduler fault tolerant agents are linked. The agentless computers are only connected to the network when it is time to run a job.

No Symphony™ file, events, or Tivoli Workload Scheduler messages are transmitted to the agentless computer. TWS consumes very little bandwidth on the network or on the agentless computer.

Disadvantages

- Not a good solution for Windows®
- Performance and latency
- Zero fault tolerance
- Extra non-TWS user management
- Potential security hole

Using the agentless method to schedule jobs does have a few disadvantages compared to a Tivoli Workload Scheduler fault tolerant agent.

Agentless scheduling is not available natively with Windows. To use the agentless method for Windows, you must install software that allows ssh logons to the Windows environment.

The agentless method makes several connections to the agentless computer every time a job is run on that computer. If there are a large number of jobs to run, the overhead and latency needed to initiate and persist those connections for each job can become burdensome.

The agentless method has no fault tolerance built in. If the network connection to the agentless computer is lost, no jobs can be scheduled there. If the network connection to the computer is lost while jobs are running, the status and output of those jobs might be lost.

Running jobs on remote computers automatically requires a remote logon connection for every user who runs jobs on the agentless computer. Each of those users must either exist on the Tivoli Workload Scheduler hosting agent or be given an ssh key to a single user. Details of this configuration are covered later.

To schedule jobs on a remote computer, the Tivoli Workload Scheduler hosting agent must be able to logon and initiate a command or script without requiring a password prompt. Typically, an ssh key without a pass phrase is needed. More secure methods are available, but these methods are complicated and cumbersome.

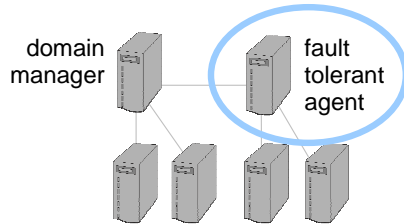
Section

Tivoli Workload Scheduler network with agentless workstations



The following section describes the Tivoli Workload Scheduler network and agentless workstations.

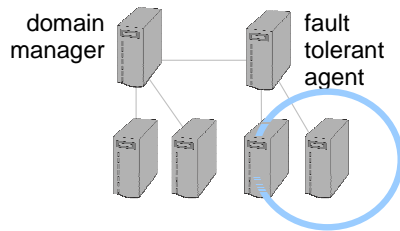
Hosting agent



- At least one fault tolerant agent needed
- This hosting agent manages the dependencies for the agentless computer
- Hosting agent must be able to connect to the agentless computer

To use agentless scheduling, at least one Tivoli Workload Scheduler fault tolerant agent is running. The fault tolerant agent can be the master domain manager or any other Tivoli Workload Scheduler agent in the network. This agent is called the hosting agent because it hosts, or manages, the remote jobs on agentless computers. More than one hosting agent can be in a Tivoli Workload Scheduler network, but these agents must be able to connect to the agentless computers.

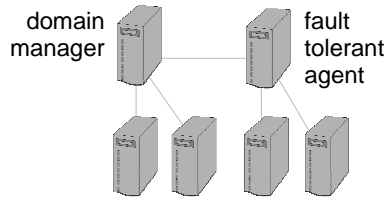
Agentless computer



- Runs jobs by connecting through the hosting agent using ssh
- Can run commands or scripts
- No software installation required on the computer

The agentless computer is a computer that can run jobs by connecting through ssh and running commands or scripts. It is not necessary to install any Tivoli software on the computer.

Preparing the computers



- User environments on the hosting agent for remote logins
- ssh and passwordless authentication are required

Now you can prepare the computers to run jobs remotely on a computer without a Tivoli Workload Scheduler agent. You must prepare the user environment for remote logins using ssh and test passwordless authentication before attempting to schedule work with Tivoli Workload Scheduler.

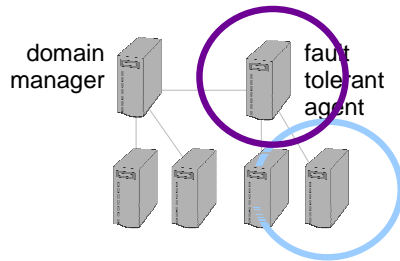
Install OpenSSH



- Most modern UNIX and Linux computers already have SSH
- Some form of SSH is required
- The installation method varies by operating system

Most modern versions of Unix and Linux already supply some form of secure shell. If yours does not, OpenSSH is a good alternative that is available for most forms of Unix. The installation of SSH or OpenSSH varies according to the operating system and is outside the scope of this module.

Determine users for launching jobs



muser = muser

twsuser = muser

- Tivoli Workload Scheduler impersonates the user
- Tivoli Workload Scheduler agent must log on without being prompted for a password
- Users on all of the agentless computers must be mapped to users on the hosting Tivoli Workload Scheduler agent computer



Every job scheduled in Tivoli Workload Scheduler is executed on behalf of a user on the computer. This situation also applies to jobs launched using the agentless method. Before defining jobs, you must prepare the environments for the users so that the Tivoli Workload Scheduler agent can log on to the agentless computer without prompting for a password. Users on all of the agentless computers must be mapped to users on the hosting Tivoli Workload Scheduler agent computer. There are two ways to accomplish this.

Match users on Tivoli Workload Scheduler and remote computer

- One-to-one...
- Create the same users on the Workload Scheduler hosting agent
- Might already be available in a network directory
- The ssh x-agent runs on the hosting agent with the same user ID as the job running remotely

The first way to map users on the Tivoli Workload Scheduler agent host to the users on the agentless computers is to create the same users on the Workload Scheduler hosting agent. This mapping might already be available in your environment if network directories such as LDAP or YP are in use. The Tivoli Workload Scheduler fault tolerant agent will run its agentless method as the user defined in the job definitions to manage the remote job.

Use the twsuser on the Tivoli Workload Scheduler computer

- The ssh x-agent runs on the hosting agent as twsuser or another user on the system
- /opt/IBM/TWA/TWS/methods/unixssh.opts
 - ▶ LJuser=twsuser
 - ▶ CFuser=twsuser
- Create all the ssh keys required for the tws user or the user you choose to log on to the remote computers. The method for creating keys is covered later in this module.

The second way to manage the user mapping is to manage all jobs running remotely by a single user on the Tivoli Workload Scheduler hosting agent. This user can be the tws user or another user on the system. You accomplish this mapping in the following steps:

1. Configure the agent.

2. Create a file called unixssh.opts in the methods directory under the Tivoli Workload Scheduler, which is the same directory where the unixssh method resides. In that file, include the following lines:

- LJ user=tws user
- CF user=tws user

3. Replace tws user with a user who is local to the Tivoli Workload Scheduler hosting agent.

4. Create all the ssh keys required for the tws user or the user you choose to log on to the remote computers. The method for creating keys is covered later in this module.

Prepare SSH keys for passwordless login

- Tivoli Workload Scheduler jobs must run completely unattended
- When Tivoli Workload Scheduler runs jobs, there is no connection to a terminal
- Tivoli Workload Scheduler as the hosting agent must be able to connect and execute commands on the agentless computers without being prompted for a password

Next, prepare SSH keys for logging in to the remote computer without a password. To execute jobs on remote computers, Tivoli Workload Scheduler must connect and execute commands on the agentless computers without being prompted for a password or being required to message to accept the remote computer as a known host. When Tivoli Workload Scheduler runs jobs, those jobs do not have any connection to a terminal and must run completely unattended.

ssh-keygen

- Use ssh-keygen to create keys for use with SSH for each user on the remote agentless computers
- For example:

```
[twsuser@twsdemo ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/twsuser/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/twsuser/.ssh/id_rsa.
Your public key has been saved in /home/twsuser/.ssh/id_rsa.pub.
The key fingerprint is:
98:59:51:69:52:6c:a2:7d:f5:db:8d:f7:64:ff:ce:0a twsuser@twsdemo.ibm.com
[twsuser@twsdemo ~]$
```

The **ssh-keygen** command creates keys for use with SSH. For each user on the remote agentless computers you must create an ssh key that does not require a passphrase to connect. For example, run the **ssh-keygen -t rsa** command. When prompted for a passphrase, press enter without entering any text. The public and private keys are created in the directory shown by the command.

Copy public keys

- Two new files are created in the \$HOME/.ssh directory
- id_rsa is the private key that resides on the hosting agent
- id_rsa.pub is the public key
- Copy the contents of id_rsa.pub to the remote (agentless) computer (appended) into \$HOME/.ssh/authorized_keys2

After you complete the **ssh-keygen** command, you will have two new files in the **.ssh** directory. The Private Key, **id_rsa**, resides on the hosting agent. The public key is **id_rsa.pub**. The public key should be copied to the remote agentless computer into the **.ssh** directory under the home directory for each user and appended to the **authorized_keys2** file.

Run a test

- Test to be sure a command can run remotely without prompting
- The first time you connect to remote computer you will receive a request for authority to connect

```
ssh twsuser@twsdemo.ibm.com who
The authenticity of host 'twsdemo.ibm.com (10.10.11.12)' can't be established.
RSA key fingerprint is 10:00:f7:d1:1c:b7:3c:64:2a:b0:2f:b1:16:3b:42:cb.
Are you sure you want to continue connecting (yes/no)? yes

ssh twsuser@twsdemo.ibm.com who
twsuser pts/0 2009-03-24 (:0.0)
```

After the keys are created and properly stored, you can run a command on the remote computer from the Tivoli Workload Scheduler hosting agent computer without receiving a password prompt. The first time you connect to a remote computer you will receive a request for authority to connect and see the prompt, Are you sure you want to continue connecting (yes/no).

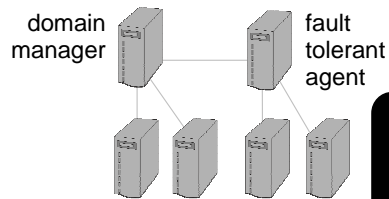
Answering yes stores the fingerprint of the remote computer into the **.ssh/known_hosts** file of the local computers. Subsequent commands should not prompt for information.

Section

Configuring Tivoli Workload Scheduler

After you have completed the configuration necessary to launch jobs remotely, you can configure Tivoli Workload Scheduler objects that represent the remote workloads you want to schedule.

Create a workstation definition for each remote computer



```
CPUNAME DEMOSSH
DESCRIPTION "Agentless TWS"
OS OTHER
NODE remote.demo.ibm.com TCPADDR 31111
TIMEZONE America/Chicago
FOR MAESTRO HOST TWSAGENT ACCESS "unixssh"
TYPE X-AGENT
AUTOLINK OFF
BEHINDFIREWALL OFF
FULLSTATUS OFF
END
```

Configure each computer that runs jobs using the agentless method as a Tivoli Workload Scheduler workstation. The workstation will be defined similar to the example on the slide.

The workstation name can be any name that fits the characteristics of a workstation, for example, 16 characters, starting with a letter, and so on.

The Operating System field must be other.

The node name is the DNS node name of the remote host.

The access keyword should point to **unixssh**.

Type is set to **X-AGENT**.

The other fields are optional.

unixssh method optional customization

Specify the managing user as the LUser setting in unixssh.opts on the hosting agent, which is typically in the /opt/IBM/TWA/TWS/methods directory

```
LUser=twuser
```

If the users on the Tivoli Workload Scheduler hosting agent do not match the users on the remote agentless computer, you can use another user on the hosting agent to manage the jobs for the agentless computer. Specify the managing user, typically the tws user, as the LUser setting in a file called **unixssh.opts**. This file resides in the same directory as the unixssh access method on the hosting agent.

Caveats

- `unixssh.opts` applies to all jobs that use the `unixssh` method
- You can make a copy of the `unixssh` method with a different name and matching `opts` file and create a different workstation that uses the new agent method

Notice that if you use the single user for managing remote jobs, the configuration applies to all jobs that use the `unixssh` method. If you want to manage some jobs using the `twuser`, and other jobs using other users, make a copy of the `unixssh` method with a different name (and thus a different options file). Then, create a different workstation that uses the copied agent method.

Section

Running jobs

After you define the workstations in Tivoli Workload Scheduler, you can build job and job stream definitions that take advantage of the remote shell execution facilitated by the agentless method.

Scheduled like any other UNIX/Linux job

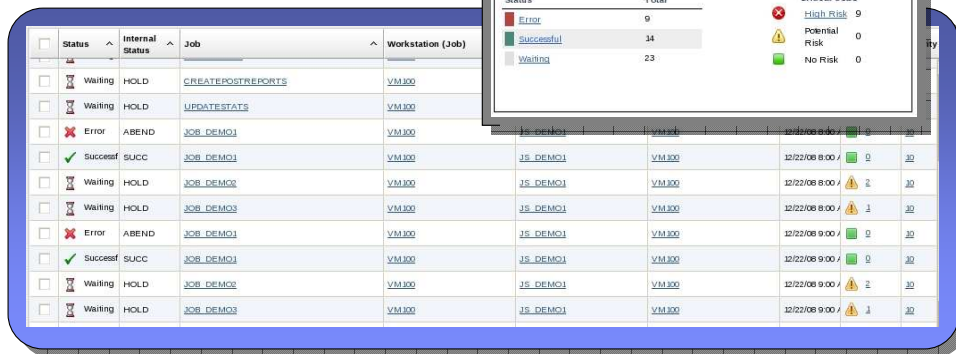
- Job definitions for agentless workstation are just like any regular UNIX job or job stream
- The stream logon must be one of the logons prepared without a passphrase

```
DEMOSSH#JOB_A
DOCOMMAND "ls -l /tmp"
STREAMLOGON remoteuser
DESCRIPTION "List files in the tmp directory"
TASKTYPE UNIX
RECOVERY STOP
These jobs can be part of any job stream.
```

Job definitions for agentless workstations are like other UNIX jobs or job streams. Each job definition contains at least a job name, a script or command to run, and a stream logon or logon user. The stream logon must be one of the logons prepared without a passphrase. These jobs can be part of any job stream.

Executing jobs

Monitoring is the same as other jobs in Tivoli Workload Scheduler



From a monitoring and management perspective, jobs scheduled for the agentless computers will look the same as other jobs in the Tivoli Workload Scheduler. You will see jobs for the agentless computer enter several states.

Agentless job states

- WAIT: The unixssh access method making the connection to the remote computer
- First time only: Transmit the wrapper script
- EXEC: Job command is running

When the dependencies of the job are satisfied, they will enter the WAIT state while the unixssh access method makes a connection to the remote computer. If no agentless jobs have run on that computer previously, the hosting agent transmits a wrapper script into **/tmp** directory of the remote computer for running jobs and capturing job output.

While the command is executing, the job should be seen in EXEC state. In some cases the WAIT state might persist.

Debugging agentless jobs

- Observe the TWS stdlist
~twuser/stdlist/yyyy.mm.dd directory
- Error output is in the ~twuser/stdlist/logs directory
- Add debug code to the unixssh method
- Redirect all output to a text file
- `set +x ;exec 2> /tmp/output.txt`

26

Agentless scheduling with Tivoli Workload Scheduler

© 2009 IBM Corporation

If your agentless jobs are not executing, or abend without running, you can take several steps to debug the problem. For these steps you should log on as the twuser.

You can find output from the jobs run by Tivoli Workload Scheduler are found in the TWS standard list directory. The name of the files contain the job number and the starting timestamp. The error output from the agent method resides in the TWS standard list logs directory. You can examine the contents of the file named with today's date for debugging information.

Because the unixssh access method is a shell script, you might want to add some code to the script to have the method provide more debugging information. For example, at the beginning of the script, replace the first few lines to make the script echo debugging information into a file named **/tmp/output.txt**. Then rerun a failed job and check the output file for more information.

For example, errors produced in the output.txt file might show that the remote connection is denied, meaning the ssh key is incorrectly configured. Or, errors produced might show that the ssh connection is prompting the user for the known hosts fingerprint and cannot be answered because the job is not running in a terminal.

Remember to remove the debugging code after you have solved the issues and have computers working.

Other management functions

- Agentless workstations cannot be linked or unlinked, stopped or started
- Those actions follow the function of the hosting agent
- Limit and fence commands are available



Agentless workstations cannot be linked or unlinked, stopped or started. For those actions the workstation follows the function of the hosting agent. You can use limit and fence commands with the agentless workstations.

Section

Customizations

When a job executes, the value of a variable is substituted with a value from the database.

Customizations

- Reduce the number of connections
- Limit the authorization of rsa keys
- Enhance or eliminate the wrapper script
- Use ssh-agent or keyring to eliminate exposure caused by empty passphrases

You can customize the agentless method to meet your unique requirements. To make customizations, copy the `unixssh` script in the `tw` user's `methods` directory to a new file with a different name. The name of the file should not be longer than eight characters. Now you can edit the new script and make custom changes.

Here are some ideas for possible customization:

The standard `unixssh` method makes several `ssh` connections to the agentless computer for every job that runs. Some of those connections can be eliminated to make the job execution more efficient. For example, if you copy the wrapper script to every remote computer manually, the `unixssh` method does not need to check for the wrapper or copy it to the remote computer. This copy process eliminates one or two connections per job.

You can customize the `ssh` authorized keys to have connections that run only a single command (perhaps the `unixssh` wrapper script) from a single IP address (perhaps the hosting agent). This customization would serve to make the `ssh` key passwordless but only allow access to a single command.

Any `ssh` logons with no passphrases are potentially unsecured. However, you can make a secure batch execution environment by using a living `ssh` agent that holds the logon `ssh` key in memory for an extended period of time. An interactive user would enter the passphrase into an existing keychain that would be managed for a defined period of time.

Section

Summary



Summary

- What is Agentless
- Agent versus Host
- User mapping
- Passwordless logins
- Workload Scheduler configurations
- Scheduling and executing
- Customizations

In this session, you have learned how IBM Tivoli Workload Scheduler can run work on remote computers that do not have Tivoli software installed. You saw the advantages and disadvantages of using this methodology over a regular Tivoli Workload Scheduler agent. You also learned how to prepare remote computers for agentless scheduling and how to set up scheduling objects and execute them. Also in the lessons were some potential customizations for streamlining agentless scheduling and making it more secure.

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

Symphony Tivoli

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2009. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

