TIDAL AGENT FOR UNIX VERSION 3.0

Cisco/Tidal provides Java-based agents for Unix and various other platforms. These Release Notes apply to the Tidal Agent for Unix Version 3.0. If you encounter any problems or have any questions, contact Tidal Software Technical Services at (650) 475-4600.

Supported Scheduler Releases

Tidal Agent for Unix Version 3.0 is compatible with the following releases of Tidal Enterprise Scheduler.

- 5.2.2
- 5.3.0
- 5.3.1

Although the agent is backward compatible with earlier versions of Scheduler, all agent functionality is not available to each release. For example, the agent supports FTPS but the FTPS feature is only accessible in Scheduler 5.3.1 and not prior releases.

New Features

The following new features are included in this release of Tidal Agent for Unix 3.0 .

FTP Client Subsystem Replaced

The FTP Client subsystem has been replaced with an entirely new implementation.

New Parameters

The following parameters have been added to support/control the new functionality (specified in *bin/tagent.ini* file):

- **SSLVLDCRT=Y/N** – Specifies whether to perform Host certificate validation on SFTP and FTPS connections. Y (yes) is the default. This is a change from the default operation of the previous agent as it did not do Host certificate validation by default. For equivalent functionality with old agent, specify **SSLVLDCRT=N** in *tagent.ini* file.
- **SSLVLDHST=<location of file containing host certification key file>** – For FTPS Host validation, the location of the file containing the public host certificates (generally self-signed), if not authenticated through a Certificate Authority.

  The certificates in the file must be of the OpenSSL PEM format and be bracketed as follows:

```
-----BEGIN CERTIFICATE-----
... first certificate ...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
```

```
... second certificate ...
-----END CERTIFICATE-----
etc
```

- **SSHVLDHST=<location of SSH host key file>** – For SFTP Host validation, the location of the file containing the public Keys for the servers that SFTP connections will be established with.

  Provides a list of hosts and their associated public keys in the given file. The format of the file is similar to that used in OpenSSH. Each line contains the name of a host followed by its IP address (separated by a comma), the type of key it has, and its key (in base-64 printable form).

  For example:

  jackspc,192.168.1.1 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIE...

- **EncryptOnly=Y|N** – The **EncryptOnly** startup parameter option has been added. **EncryptOnly=Y** will cause an Agent to not remain connected to any Master that has turned off message encryption (it is on by default).

Security Enhancements

Masters.cfg file at the Agent

In order to provide strict control over which Tidal Enterprise Scheduler masters can connect to a specific agent, a *Masters.cfg* file has been implemented at the Agent. By specifying the Master 'alias', the Master 'alias' and a specific 'local' TCP/IP address or the Master 'alias', the specific 'local' TCP/IP address and a 'global' TCP/IP address you can uniquely identify the specific Enterprise Scheduler Masters that a Agent will create connections to.

The *Masters.cfg* file must be created in the Agents local directory. This directory is in the install path of the Agent and has the name of the Agent as it was specified when the Agent was defined. For example, by default, this would be something like:

- Unix (Linux, z/OS)

  /opt/TIDAL/Agent/TidalAgent1

- Windows

  C:\Program Files\TIDAL\Agent\TIDAL_AGENT_1

- OVMS

  sys$sysdevice:[tidal.agent.tidalagent1]

This file should have limited access using native system access control definitions.

Agent Connect Protocol

The following describes the normal connection sequence for an Agent to Master connection to be established.

The Master connects to the Agent well-known port (default **5912**, configurable). The Master sends a registration message to the Agent specifying the Masters IP address and

listening port (and some other configuration information). This connection is then terminated.

For each Master that has registered as above, the Agent will attempt to connect using the information from the registration. This will happen each time the connection is lost for any reason.

The Agent will attempt to connect to the IP and port provided by the Master in the registration message. If this fails, the Agent will attempt to connect to the IP obtained from the network as the source IP (may be firewall IP) and the port provided in the registration message.

When the connection is made, the Agent will generate an encryption key based on a random seed. This encryption key and other configuration information about the Agent will be sent to the Master. The encryption key is 'wrapped' by a method that the Master knows how to 'unwrap' in order to get the raw key. This key is used to encrypt the body of all future messages (encryption is a configurable option that is on by default).

Masters.cfg

The *Masters.cfg* file has the following structure:

- Any line beginning with '#' is considered a comment and ignored.
- Optional **`INCLUDE`** or **`EXCLUDE`** statement on first (non-comment) line. If specified, these one word entries must be on the first (non-comment) line. **`INCLUDE`** is the default if nothing is specified.
  - **`INCLUDE`** – only the specified Masters with optionally specified IP addresses will be connected to by the Agent.
  - **`EXCLUDE`** – the specified Masters will be specifically excluded from being connected to by the Agent.
- Master entries of the form:

  `MasterAlias`

  `MasterAlias:IPaddress1`

  `MasterAlias:IPaddress1;IPaddress2`

  `MasterAlias`

  The **`Masteralias`** comes from the **`Machine Name`** field on the Master Connection.

`MasterAlias`

The **`Masteralias`** comes from the **`Machine Name`** field on the Master Connection definition (see Figure 1 below). The comparison is case-insensitive.

If specified alone on the line, then only the **`MasterAlias`** will be verified that it matches what was presented by the Master in its registration message.

`IPaddress1`

For connections that are 'local', i.e. their Master host machine IP addresses are directly accessible by the Agent, then only **`IPaddress1`** needs to be specified. This address will be verified against the IP address presented by the Master in its registration message and the IP address obtained from the network as the origination of the connection that provided the registration message.

IPAddress2

For connections that must traverse a firewall, then **IPaddress2** must be specified. **IPaddress2** will be the externally known address of the firewall. The externally known address of the firewall is what will be obtained by the Agent when it retrieves the IP address of the origination of the connection through which the registration message was delivered.

For situations where a Master could have multiple IPs, Failover scenarios, or disaster recovery situations, the same **MasterAlias** can be specified with different IP address parameters.

An example of a *Masters.cfg* file:

```
INCLUDE
hou-testvm-531:192.168.48.111;172.19.25.125
hou-testvm-531:192.168.55.211;172.19.25.225
HOU-PROD:192.168.95.92
HOU-PROD:192.168.42.92
catest
```

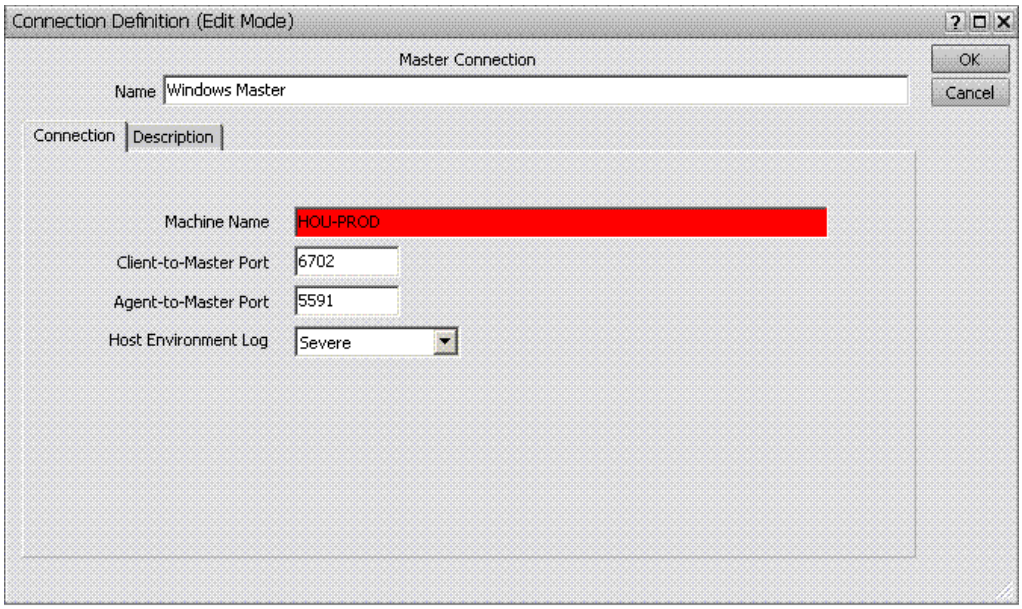

Figure 1 Master Connection Definition Dialog (MasterAlias in Red)

Bugs Fixed in 3.0

BUG00520

Events cannot move files with file names that have spaces.

Compatibility

Table 1 Compatibility Matrix

OS Name	Version	Chipset	32 bit	64 bit	JVM
HPUX	11.11	PA-RISC	X		HP 1.6.0
HPUX	11.223	Itanium		X	HP 1.6.0
AIX	6.1	PowerPC/ RISC	X	X	IBM 1.5.0
AIX	5.3	PowerPC/ RISC	X	X	IBM 1.5.0
Solaris	9	Sparc	X	X	Sun 1.5.0
Solaris	10	Sparc	X	X	Sun 1.5.0
Solaris	10	Opteron		X	
Linux	Redhat Enterprise Linux AS Release 4 & 5	Intel/AMD	X	X	Sun 1.5.0
Linux	SUSE Enterprise Server v8, v9, v10	Intel/AMD	X	X	Sun 1.5.0
Linux	Oracle Enterprise Linux 5.2	Intel/AMD	X	X	Sun 1.5.0
Linux	openSUSE 10.2 (i586) - Kernel 2.6.18.8-0.9-default	Intel/AMD	X	X	Sun 1.5.0
Linux	Linux Kernel 269 or above	PowerPC	X	X	IBM Java 1.5
Tru64	v5.0A or above	Alpha		X	HP 1.4.2-5
VMWare	ESX 2.5, ESX 3.0, ESXi 3.5				
Microsoft Virtual Server	2005				
Cent OS/ SCO	Call				