HOW TO CONFIGURE MODEL MANAGER TO WORK ON SSH TUNNEL

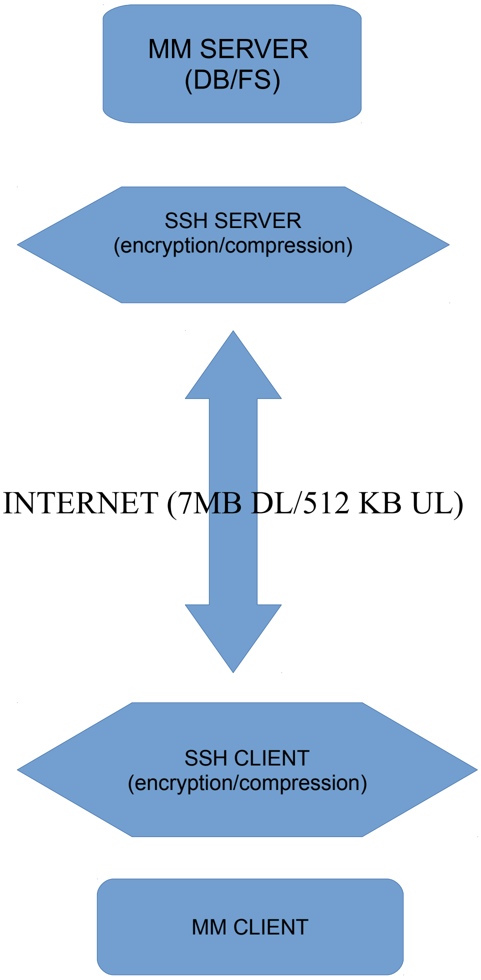
The scope of this paper is to show an alternative way to connect Model Manager client and server, when normal dsl lines (7MB DL, 512 kB UL) are used. The scenario is: the server is located in the "customer" plant, the client is located in the "supplier" office, communications pass through the internet.
Using an SSH tunnel, we have two benefits: data encryption and data compression. The last one is really important to speed-up data exchange, due to the fact that MM infrastructure does not compress data before sending.

The big part of exchanged data are:

1) sql-queries (text)
2) model infos (text)
3) models previews (jpg)
4) geometry files 3d/2d (3d-data or mi format)

About 1) and 2) we could expect a reduction of weight till 10% of the original size, while 3d-data can be compressed till 25% of the original size (try to zip a 3d-data folder: you will find a final weight that will be around the same of the corresponding pkg file).

By inserting an SSH tunnel between client and server, we obtain this schema:



USED SOFTWARE:

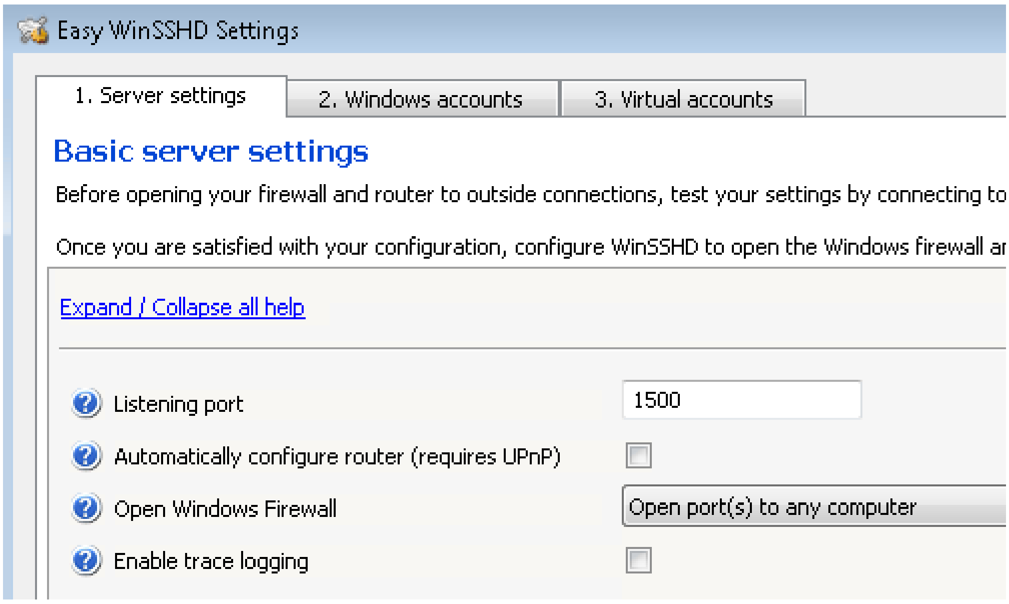
1) SSH server for windows: winsshd by bitvise http://www.bitvise.com/winsshd
2) SSH client for windows: putty http://www.chiark.greenend.org.uk/~sgtatham/putty/

CONFIGURATIONS:

SERVER SIDE:

SSH Server must be installed on a computer inside the same “customer” LAN where the model manager server is executed. The SSH server may run on the same machine where model manager is running, but this is not mandatory: the only difference is in the client configuration.
NOTE: I experienced authentication problems when the SSH server was first installed on a machine that hosted the windows domain server on the customer LAN, so it is better to choose a different computer.

Winsshd installation is really easy, just choose a “personal profile” (this makes it free, no cost). The only special settings are:
a) Set the listening port other than 22 standard port, for security reasons. I choose the 1500 port for this example.
b) Create a real windows user on the ssh server host, and then associate it to the users allowed to connect remotely to the ssh server.



Easy WinSSHD Settings

Basic server settings

Before opening your firewall and router to outside connections, test your settings by connecting to

Once you are satisfied with your configuration, configure WinSSHD to open the Windows firewall ar

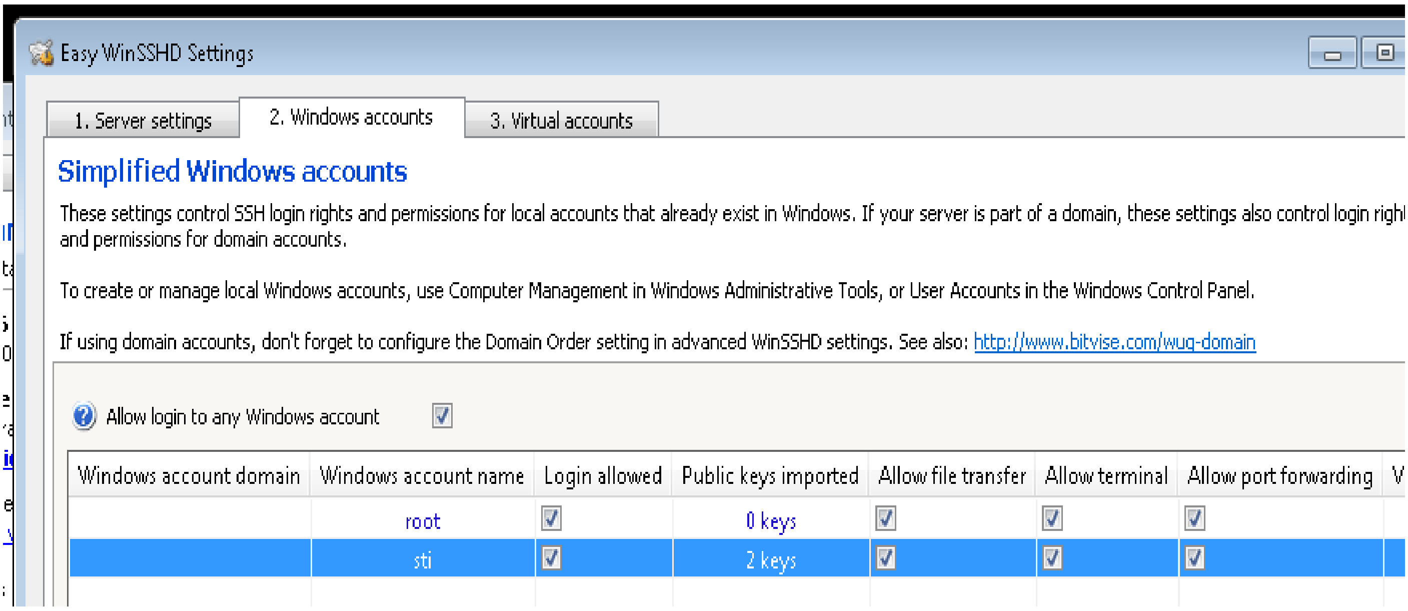
Expand / Collapse all help

Listening port: 1500

Automatically configure router (requires UPnP) □

Open Windows Firewall: Open port(s) to any computer

Enable trace logging □



Easy WinSSHD Settings

Simplified Windows accounts

These settings control SSH login rights and permissions for local accounts that already exist in Windows. If your server is part of a domain, these settings also control login righ and permissions for domain accounts.

To create or manage local Windows accounts, use Computer Management in Windows Administrative Tools, or User Accounts in the Windows Control Panel.

If using domain accounts, don't forget to configure the Domain Order setting in advanced WinSSHD settings. See also: http://www.bitvise.com/wug-domain

Allow login to any Windows account ☑

Windows account domain	Windows account name	Login allowed	Public keys imported	Allow file transfer	Allow terminal	Allow port forwarding	V
	root	☑	0 keys	☑	☑	☑	
	sti	☑	2 keys	☑	☑	☑	

The winsshd installer configures the ssh server to be executed as service, so it is not necessary to start the service manually.
Last: the "customer" router must be configured to forward the 1500 port to the local ip of the SSH server host.

CLIENT SIDE:

Here the putty client config is shown: an user profile is created, with following options:

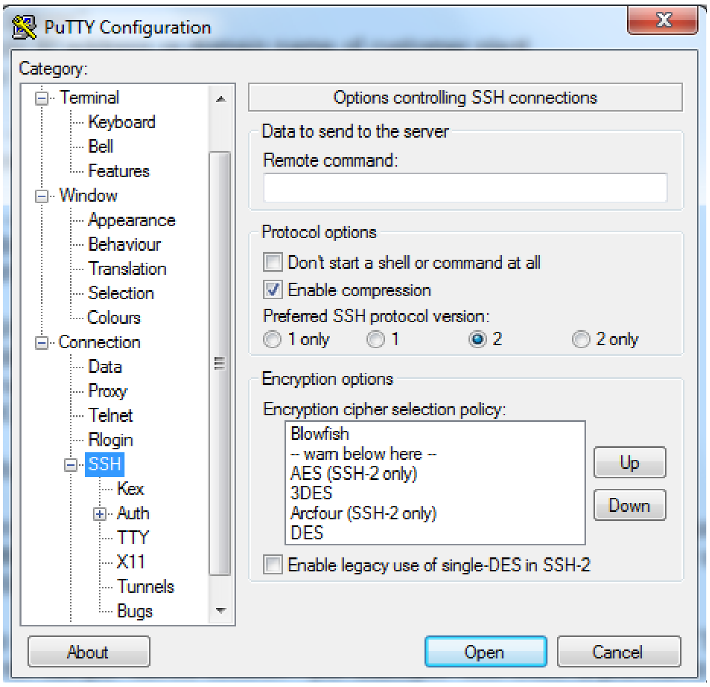
1) Main page "Basic Options":
   1. Host name: **<u>public</u>** IP address or domain name of customer plant
   2. SSH port where the server is listening on

2) Connections Page → data Menu → auto-login username
   1. Insert the windows username that was created on the ssh server host

3) Connections page → SSH Menu
   1. mark the checkbox "Enable compression"
   2. choose "blowfish" as ecnryption algorithm (it is the lighter)

4) Connections → SSH Menu → Tunnels Menu
   1. source port: 9898
   2. Destination: server_local_ip_MM:9898*
   3. Destination: LOCAL
   4. ADD
   5. redo the same for 9899 (and other ports/services, if needed)

- * = If the SSH server is executed on the same machine where MM server is running, the "server_local_ip_MM" will be 127.0.0.1, otherwise we will write the internal local IP on the MM server, example: 192.168.0.2



Options controlling SSH connections

Data to send to the server

Remote command:

Protocol options

□ Don't start a shell or command at all

☑ Enable compression

Preferred SSH protocol version:

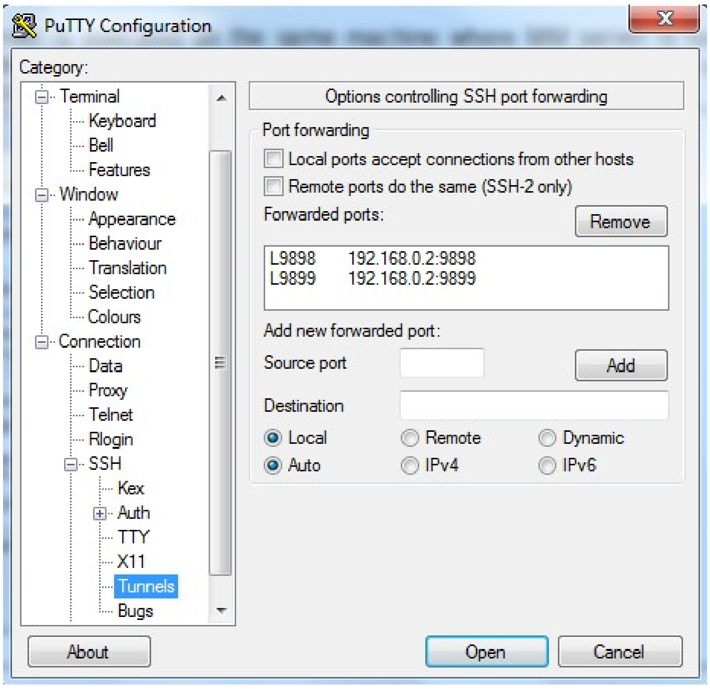
○ 1 only ○ 1 ● 2 ○ 2 only

Encryption options

Encryption cipher selection policy:

Blowfish
-- warn below here --
AES (SSH-2 only)
3DES
Arcfour (SSH-2 only)
DES

□ Enable legacy use of single-DES in SSH-2



Options controlling SSH port forwarding

Port forwarding

□ Local ports accept connections from other hosts

□ Remote ports do the same (SSH-2 only)

Forwarded ports:

```
L9898     192.168.0.2:9898
L9899     192.168.0.2:9899
```

Add new forwarded port:

Source port

Destination

● Local ○ Remote ○ Dynamic

● Auto ○ IPv4 ○ IPv6

Now, it is possible to store these infos in a user profile:
in the first page of the putty GUI, write the desired name of the profile (example: "Customer-SSH") in the area "saved sessions", then press the button "save".

The profile is stored in the HKCU key:

HKEY_CURRENT_USER\Software\SimonTatham\PuTTY\Sessions\Customer-SSH

MODEL MANAGER CLIENT CONFIGURATION

- In order to be able to enroute the client queries through the SSH channel, we need to create a new model manager user profile.

- Let's go to the folder: C:\Users\STI\AppData\Roaming\CoCreate\Model Manager\17.0, and copy and paste the existing custom.xml profile to (example) “customer-SSH.xml”.

- Open the new created xml profile and modify it as follow (wordpad is a good tool to do this):
  - <DbServer>127.0.0.1:9898</DbServer>
  - <FileServer>127.0.0.1:9899</FileServer>
  - We can add here other services: software distribution server, license server, etc etc.

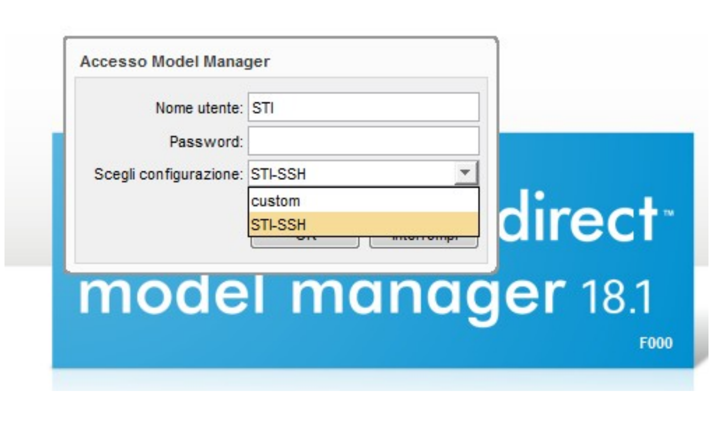
    - These settings will make that all queries directed to localhost:9898/9899/etc will be intercepted by putty, and redirected to the real server on the customer remote LAN.

DAILY USE:

It is necessary to establish a ssh connection before launch OSD and MM:

Putty → Load → Customer-SSH → Open

A cmd window will popup and we will insert username (if needed) and password. After the connection will be successfull, we will launch OSD and MM: when more than one xml user profile exists, MM asks which one we want to use:


Accesso Model Manager

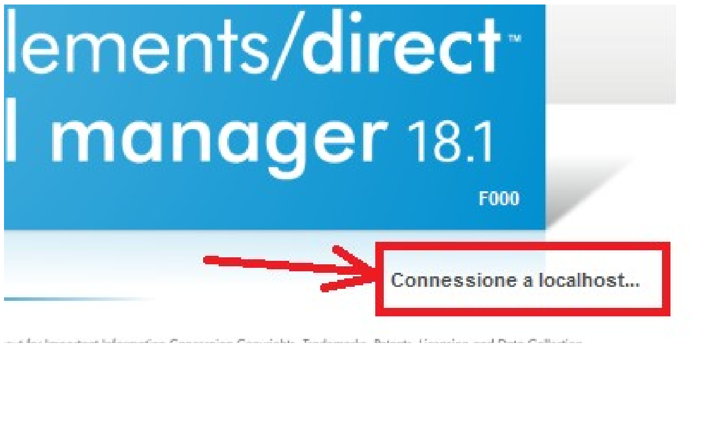
Nome utente: STI

Password:

Scegli configurazione: STI-SSH

custom

STI-SSH


Connessione a localhost...

RESULTS:

the scope of the use of an SSH tunnel was to save time due to data compression, and a good performance was obtained in a quick test: the time needed to load an assembly of 150 unique components and total size (3d-data format) of 10.6 MB was reduced from 230 seconds to 90 seconds.

Please feel free to make comments, questions and suggestions:

cacciatorino@libero.it