



Session Saver

Georgia SoftWorks has again pioneered another outstanding feature for Windows 7/NT/XP/VISTA/2000/2003/2008 Servers. This feature Saves the Telnet/SSH2 Session in the event of a link or client failure and allows users to reconnect to the same session the next time they log in and *resume the work in progress, exactly where they left off*. Extremely useful when connectivity is across the Internet or with Radio Frequency Barcode/Scanner applications where it is important to continue work in progress, even after a connectivity failure. If a reconnection is not performed within a specified time period then Graceful Termination will initiate. For the fastest reconnection and minimal prompting you may want to couple the Session Saver with AutoLogon (page 160). Especially useful in RF Data collection applications.

Upon a client or link failure, the SSH2/Telnet session and associated NTVDM are normally terminated eliminating any orphaned sessions. This is the default behavior. However there are times when it may be desirable to *Save the Session* giving the user the opportunity to reconnect to the session to resume work in progress. This session is termed to be *Saved* or *Suspended*¹⁷⁷. This feature can be enabled as described below.

Three methods exist to connect to a Suspended session. They are the *Session Reconnect Method* and the *Attach Method* and using Team Services Recover (page 118).

The Session Reconnect Method is defined to automatically recognize when a User or User from a specific IP address is Logging In and to determine if they have any Saved or Suspended sessions. If so, then the telnet/SSH2 server will automatically reconnect the user to their existing *Saved Session*.

The *Attach Method* is defined to allow a User that is a member of the **Gwtn Monitors** group to identify any Suspended sessions and to Attach (connect) to that session. Please see the Session Monitor for more information on this method (page 162).

The Session Reconnect method can be configured on either a global or per user basis by utilizing Logon Scripts (page 189). Two environment variables are used in configuring the Session Reconnect.

The environment variables for the Session Reconnect are:

gwtn_reconnect

Possible values are

- **auto_by_user_and_ip** - Reconnect based on User Id and IP Address
- **auto_by_user** - Reconnect based upon User Id



And

gwtm_reconnect_timeout

specifies time in minutes for how long broken sessions will wait for the original user to reconnect. This variable does not have to be set, the default value is 0. It must be set to non-zero for Session-Reconnection to operate.

Note: The User should be using the same terminal emulation during a reconnect that they were using when the failure occurred.

Session Reconnection Timeout

A system administrator may not want Suspended sessions to stay around indefinitely. The amount of time in minutes that a user's session can be available for Session Reconnection is specified by the environment variable **gwtm_reconnect_timeout**. Once this timer has expired for a user then Graceful Termination initiates.

This environment variable must to be set because the default is 0 minutes, which is disabled. This environment variable is ignored if the **gwtm_reconnect** environment variable is not set.

For example to set the Session Reconnection Time to 1 hour you would enter:

```
set gwtm_reconnect_timeout=60
```

in the Logon Script for a particular user.

NOTE: No spaces are allowed when setting environment variables.

For example:

```
set gwtm_reconnect_timeout=60 is correct
```

```
set gwtm_reconnect_timeout = 60 is not correct
```

Reconnection based on User ID.

Let's look at an example. User "Bob" is using a RF Hand Held terminal and is performing a multi-point inspection of an item. Bob is navigating through many screens, logging data during the inspection. Half



way through the inspection the Hand Held terminal fails. It would be great if Bob could simply grab a spare Hand Held Terminal, log in and continue his work in progress.

This is possible with Reconnection based on User ID. The SSH2/Telnet server will determine if there are any Suspended sessions for user Bob. If there are Suspended sessions then the SSH2/Telnet server will automatically connect Bob to the existing session. The screen will be exactly in the same condition as it was when Bob was previously connected and Bob can continue work in progress!

For example to enable Session Reconnection you would enter:

```
set gwnn_reconnect=auto_by_user
```

in the Logon Script for a particular user.

NOTE: No spaces are allowed when setting environment variables.

For example:

```
set gwnn_reconnect=auto_by_user is correct
```

```
set gwnn_reconnect = auto_by_user is not correct
```

Reconnection based on IP Address and User ID.

There may be times when you have permanent IP addresses and you only want users from machines at those locations to be able to reconnect to Suspended sessions for security or other reasons. In this situation you may consider reconnection based on IP address AND User Id.

Let's look at an example. User "Jane" is using a Medical Application from a secure location. The medical application can only be used at this specific location for security reasons. She is almost complete entering a new patient when a circuit breaker trips. It would be nice if when the circuit breaker was reset she could login and automatically continue her work in progress.

This is possible with Session Reconnection based on IP Address and User ID. The SSH2/Telnet server will determine if there are any Suspended sessions for user "Jane" from the IP address from which she is connecting. If there are Suspended sessions for that user AND that IP address then the SSH2/Telnet server will automatically connect Jane to the existing Saved session. The screen will be exactly in the same condition as it was Jane was last connected and she can continue her work in progress!

For example to enable this type of Session reconnection you would enter:

```
set gwnn_reconnect=auto_by_user_and_ip
```

in the Logon Script for a particular user.



NOTE: No spaces are allowed when setting environment variables.

For example:

`set gwtm_reconnect=auto_by_user_and_ip` is correct

`set gwtm_reconnect = auto_by_user_and_ip` is not correct