

Introduction

This text describes GMS scoreboard definitions and drivers. In addition, it documents the *Remote Serial Server*, a separate application which allows multiple copies of GMS to connect to the same device or a copy of GMS to drive a scoreboard which is not physically attached to the machine on which GMS is running.

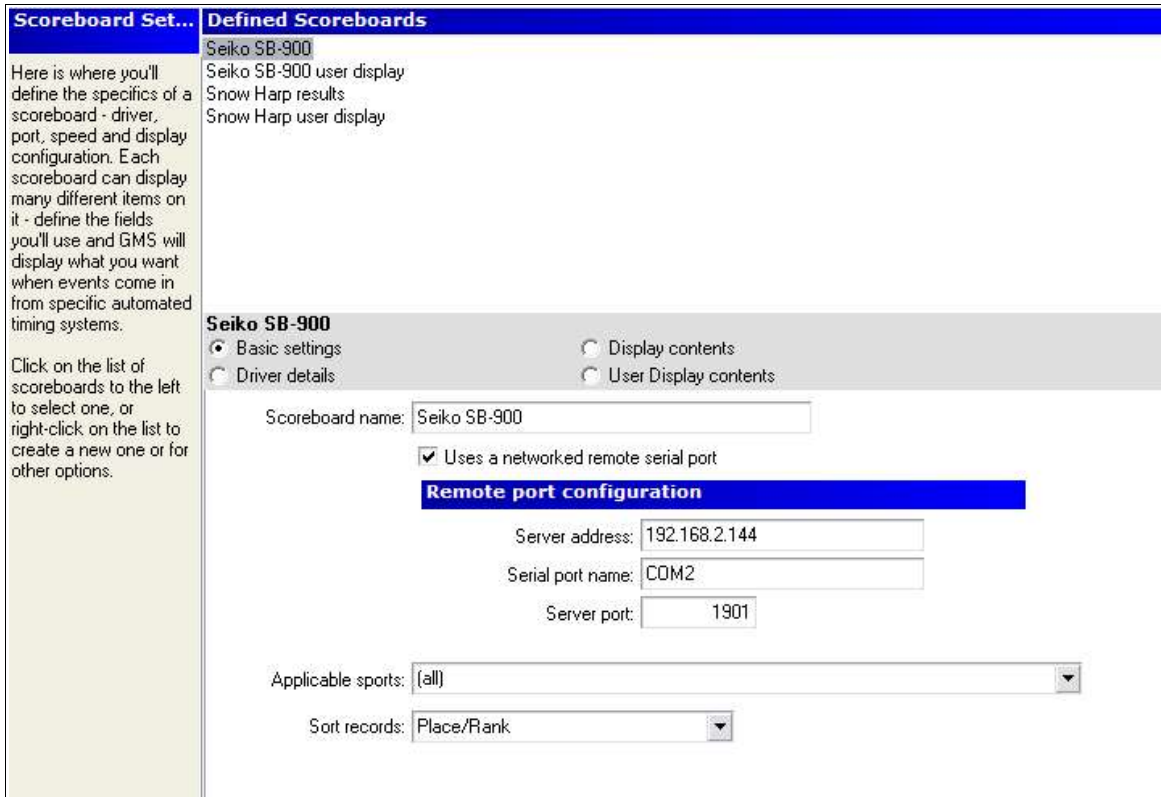


Illustration 1, Scoreboard Setup

Scoreboard Setup

Scoreboards are defined in four parts: Basic settings, Driver details, Display contents, and User Display contents.

Basic settings configures the scoreboard's name, how it connects to the physical device or remote serial port, sports to which it is applicable, and how records sent to it should be sorted.

Driver details defines the codes used to communicate with the device, the device's size and layout.

Display contents and User Display contents lay out the data to be displayed on the board. The display contents are those which are sent through the driver out to the physical scoreboard; the user display contents are shown on the internal scoreboard viewer. In many cases, these will be the same.

Context

Scoreboards added through the system "Scoreboards setup" under "System configuration" are available within all games. Scoreboards added while inside a game are only available within that one game. For systems which are primarily concerned with the running of a single game, this is immaterial; for those which are used to run many games, it is usually best to define game-specific scoreboards from within the relevant game so that they don't appear in other games where they are not applicable.

To define a new scoreboard, right-click on the list of scoreboards.



Click on “New” to define a new scoreboard from scratch, or “Load a definition from disk” to load a saved scoreboard layout (GMS scoreboards are saved with the extension “.gdd”).

Basic Settings

A screenshot of a configuration dialog box titled 'Local port configuration'. At the top, there is a text field for 'Scoreboard name:' containing 'Snow Harp results'. Below it is a checkbox labeled 'Uses a networked remote serial port' which is unchecked. The main section is titled 'Local port configuration' in a blue header. It contains several dropdown menus: 'Serial port:' set to 'COM3:', 'Port speed:' set to '9600 baud', 'Parity:' set to 'None', 'Data bits:' set to '8', and 'Stop bits:' set to '1'. Below these is another dropdown menu for 'Applicable sports:' set to 'Cross Country Skiing'. At the bottom is a dropdown menu for 'Sort records:' set to 'Do not change'.

Illustration 2, Basic scoreboard settings

This defines the scoreboard used at the Snow Harp venue in Hakuba, Japan – a complex multi-line, alphanumeric scoreboard. The important values here are the name, the fact that it's connected to a local physical serial port (COM3:), is applicable only to Cross Country Skiing events, and when records are sent to it, their order is not changed.

The same scoreboard, but defined as using a remote serial port, is shown below. This scoreboard expects a Remote Serial Server at the IP address 192.168.2.144, connects to the port labeled “SnowHarp”, and expects the server to be listening on IP port 1901 (the default port for a Remote Serial Server). (Remote serial ports are only available in GMS 5.2.0.17 and higher.)

A screenshot of a configuration dialog box titled 'Remote port configuration'. At the top, there is a text field for 'Scoreboard name:' containing 'Snow Harp results'. Below it is a checkbox labeled 'Uses a networked remote serial port' which is checked. The main section is titled 'Remote port configuration' in a blue header. It contains three text input fields: 'Server address:' with '192.168.2.144', 'Serial port name:' with 'SnowHarp', and 'Server port:' with '1901'. Below these is a dropdown menu for 'Applicable sports:' set to 'Cross Country Skiing'. At the bottom is a dropdown menu for 'Sort records:' set to 'Do not change'.

Driver details

This is the most complex portion of defining a scoreboard, and requires a detailed knowledge of what commands are used to communicate with the scoreboard and send it data. The two relevant sections here are

[Driver.Ops] and [Driver.Limits].

[Driver.Limits] defines the layout of the board and how often it can be refreshed. In the first example, it defines a board comprised of one eight segments: one across (MultiLayout=1x15), 15 characters across total (Width=15) and eight segments high (Height=8), with each segment holding ten characters (MultiLayout=1x10).

In the second, it shows a scoreboard made up of eighty segments: eight across (MultiLayout=8x5), forty characters across total (Width=40), and ten segments high (Height=10), with each segment holding five characters (MultiLayout=8x5).

[Driver.Ops] tells GMS the commands to use to send data to each of the physical segments of the board. Since the sample board here is one segment wide and eight segments high, we give commands for each of the eight units. (GMS will take care of breaking up the actual data sent into portions for each segment.)

In addition to one command per segment, there are three additional commands which can be used if necessary: Clear, FinishUpdate and StartUpdate. StartUpdate will be sent before any set of update commands to the scoreboard, FinishUpdate will be sent at the end of that set, and Clear will be used to empty the display.

Command layout

Every command is comprised of an optional prefix, the {\$!Data} command, and an optional suffix. The {\$!Data} command will be replaced with the actual data as defined by the content section of the board's definition. All other characters are sent literally, or translated into other characters ({\$xxx} characters). Values in the format {\$xx} are converted to hex – either using Table 1, or as hexadecimal values.

Symbol	Hex
NUL	00
SOH	01
STX	02
ETX	03
EOT	04
ENQ	05
ACK	06
BEL	07
BS	08
TAB	09
LF	0A
VT	0B
FF	0C
CR	0D
SO	0E
SI	0F
DLE	10
DC1	11
DC2	12
DC3	13
DC4	14
NAK	15
SYN	16
ETB	17
CAN	18
EM	19
SUB	1A
ESC	1B
FS	1C
GS	1D
RS	1E
US	1F

Table 1, These special values are recognized and converted into the corresponding hex values.

```
[Driver.Ops]
FinishUpdate=
StartUpdate=
Line1,1={$STX}C01{!Data}{$ETX}
Line2,1={$STX}C02{!Data}{$ETX}
Line3,1={$STX}C03{!Data}{$ETX}
Line4,1={$STX}C04{!Data}{$ETX}
Line5,1={$STX}C05{!Data}{$ETX}
Line6,1={$STX}C06{!Data}{$ETX}
Line7,1={$STX}C07{!Data}{$ETX}
Line8,1={$STX}C08{!Data}{$ETX}
Clear={$STX}C63 {$ETX}
```

```
[Driver.Limits]
Width=15
MultiLayout=1x15
MaxRefresh=500
Height=8
```

Text 1, Sample simple scoreboard

```
[Driver.Ops]
StartUpdate=
Line10,1={$STX}a:11{!Data}{$ETX}
Line10,2={$STX}a:21{!Data}{$ETX}
Line10,3={$STX}a:31{!Data}{$ETX}
Line10,4={$STX}a:41{!Data}{$ETX}
Line10,5={$STX}a:51{!Data}{$ETX}
Line10,6={$STX}a:61{!Data}{$ETX}
Line10,7={$STX}a:71{!Data}{$ETX}
Line10,8={$STX}a:81{!Data}{$ETX}
Line1,1={$STX}a111{!Data}{$ETX}
Line1,2={$STX}a121{!Data}{$ETX}
Line1,3={$STX}a131{!Data}{$ETX}
Line1,4={$STX}a141{!Data}{$ETX}
Line1,5={$STX}a151{!Data}{$ETX}
Line1,6={$STX}a161{!Data}{$ETX}
Line1,7={$STX}a171{!Data}{$ETX}
Line1,8={$STX}a181{!Data}{$ETX}
...(lines remove for clarity)...
Line8,5={$STX}a851{!Data}{$ETX}
Line8,6={$STX}a861{!Data}{$ETX}
Line8,7={$STX}a871{!Data}{$ETX}
Line8,8={$STX}a881{!Data}{$ETX}
Line9,1={$STX}a911{!Data}{$ETX}
Line9,2={$STX}a921{!Data}{$ETX}
Line9,3={$STX}a931{!Data}{$ETX}
Line9,4={$STX}a941{!Data}{$ETX}
Line9,5={$STX}a951{!Data}{$ETX}
Line9,6={$STX}a961{!Data}{$ETX}
Line9,7={$STX}a971{!Data}{$ETX}
Line9,8={$STX}a981{!Data}{$ETX}
Clear={$STX}C001{$ETX}
FinishUpdate={$STX}D001{$ETX}
```

```
[Driver.Limits]
Height=10
Width=40
MaxRefresh=500
MultiLayout=8x5
```

Text 2, Sample complex scoreboard values

Scoreboard content

Fields and their layouts define the contents of a scoreboard. These are independent of the driver, which defines communication with the scoreboard. Every scoreboard has two sets of contents: display contents and user display contents.

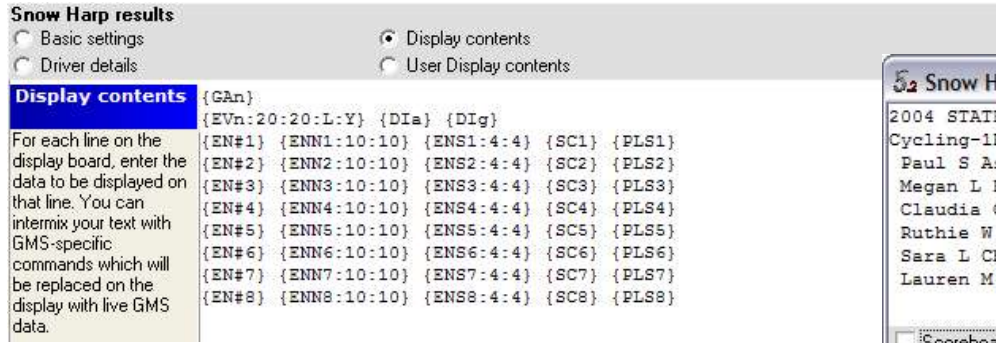


Illustration 3, Sample scoreboard contents



Illustration 4, Sample scoreboard display

Any combination of literal characters and data fields can be included in a scoreboard's contents. Reference the tables "General fields" and "Entrant-specific fields" for what data fields can be displayed.

Using "User/" fields allows scoreboards to be driven by user input, rather than just the data from a GMS form. For example, the following setup will show a scoreboard with a fixed header ("Special Olympics 2005 World Games") with nine lines of user-entered data below it:

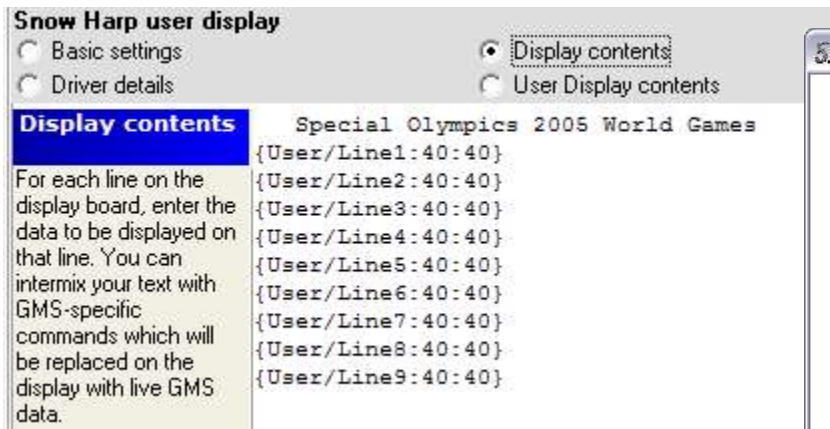


Illustration 5, Sample user display scoreboard



Illustration 6, User entry portion of scoreboard

When used within GMS, Illustration 6 is an example of how the user can interact with the scoreboard. Any text data can be entered in the given fields, and when the user clicks on "Update display", that data will be sent to the scoreboard.

General fields

Field	Description
user/display name	Provides space for user-entered text value named "display name"
day	Date in DD MMMM YYYY format
DAB	Date in GMS standard date format
DAT	Time in hh:nn format
DA1	Time in hh:nnam/pm
GAN	Games name
SPC	Sport code
SPN	Sport name
EVN	Event name or scoreboard name
EV#	Event number
RAT ***NOT IMPLEMENTED***	Current race clock time
ROC	Current round abbreviation
RON	Current round description
DIN	Current division
DIA ***NOT IMPLEMENTED***	Age groups for current division
DIG	Gender of current division
DIS	Gender code of current division
DIT	Start time for current division
DIQ	Status for current division

Entrant-specific fields

For all of these codes, put the code followed by the entrant number, e.g. "{EN#2}" for entrant 2's bib number.

Field	Description
ENN	Entrant name, e.g. "Robbins, Alfred"
ENNB	Entrant name, e.g. "Robbins, A"
ENNR	Entrant name, e.g. "Alfred Robbins"
ENNL	Entrant family name, e.g. "Robbins"
ENG	Entrant gender, e.g. "F"
ENA	Entrant age
ENB	Entrant bib #
ENS	Entrant abbreviated delegation name
END	Entrant delegation name
EN#	Entrant lane #
SS	Scratch/qualifying score
SC	Current score
PLS	Entrant place abbreviated, e.g. "1", "10", "D"
PL	Entrant place, e.g. "1st", "DQ"

Formatting

Optionally suffix all commands by formatting options, each delimited by colons.

pad width
truncation width
alignment (L, R, or C)
compress spaces (Y or N)

For example, "{ENN2:30:40:L:Y}" indicates to take the entrant's name, padded to thirty characters, truncated to forty characters, aligned left, with any runs of two or more spaces changed to a single space.

Modifiers

For all fields, the field name can be suffixed with zero or more of the following options:

\KEEPN - removes all but the numeric characters "0".."9"
\KEEPA - removes all but the alphabetic characters "A".."Z" and "a".."z"
\KEEPB - removes all but the characters "A".."Z", "a".."z" and "0".."9"
\UPPER - changes the text to all upper-case
\LOWER - changes the text to all lower-case

For example,

{DAT\RAWN} would result in "1245" if the current time was 12:45.

{ENN\UPPER\KEEPA} for an entrant whose name is "Smith, John 3" would result in "SMITH, JOHN"

Remote Serial Server

The *Remote Serial Server* is a special tool for allowing multiple copies of GMS to connect to the same device, for a copy of GMS to drive a scoreboard which is not physically attached to the machine on which GMS is running, or any combination of these.

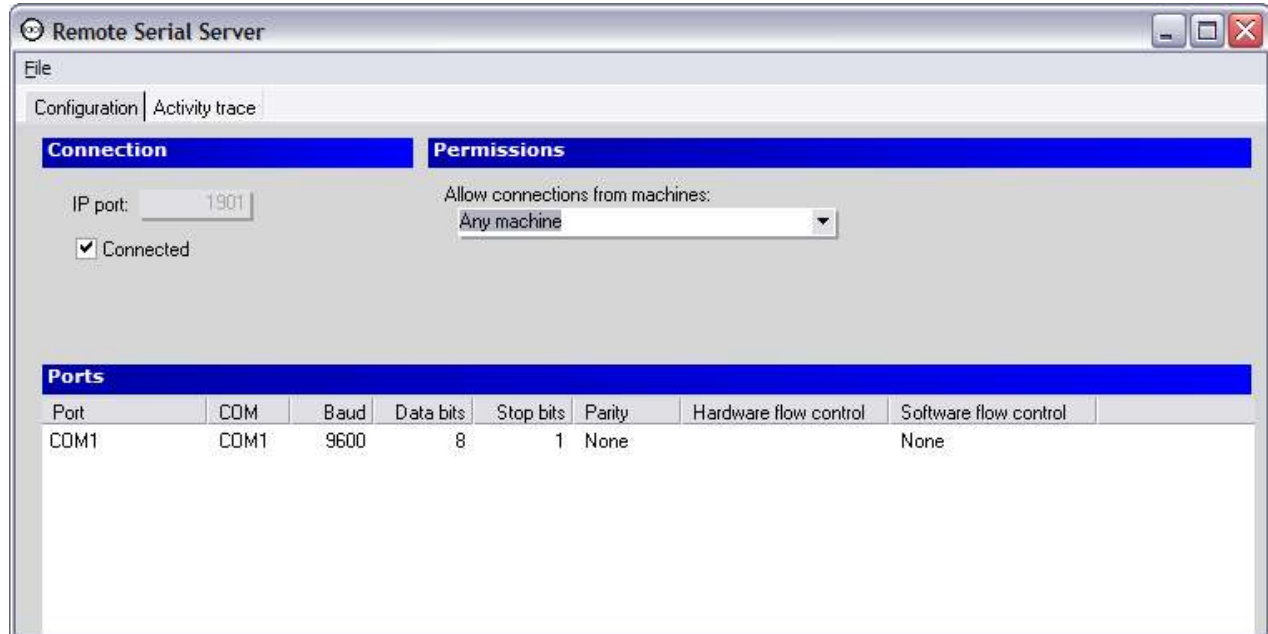


Illustration 7, Remote Serial Server

By defining one or more ports, copies of GMS can use these serial ports remotely from any machine which can see this application across the network.

Connection

By default, *Remote Serial Server* listens on port 1901 for network connections. (This port has nothing to do with the physical ports defined for sharing.) Multiple copies of *Remote Serial Server* can run on the same machine as long as they are listening on different network ports.

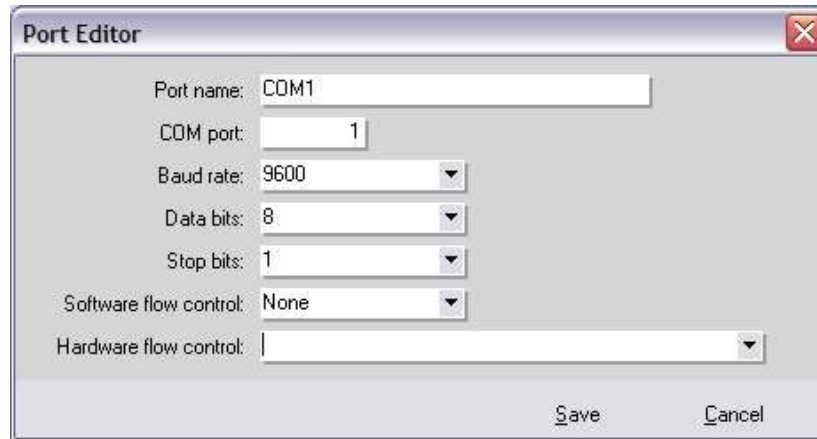
Permissions

By default, *Remote Serial Server* allows connections from anywhere. Changing the permissions value allows you to define a “blacklist” - a list of machines which are not allowed to connect, or a “whitelist” - a list of machines which is allowed to connect, and *Remote Serial Server* will enforce these restrictions for all new, incoming connections.

Ports

This is a listing of the physical ports on this machine being shared. Right-click on the list or an existing item to create a new port or change an existing port.

Note that all configuration options – port, permissions, new ports or changes to existing ports are not saved for future use unless you click on “File” than “Save settings”. Configuration information is saved in the `rsServer.ini` file.



A sample port definition is shown above. The name (in this case, “COM1”) is the name by which a remote copy of GMS will refer to this port. It is not necessary to name it in the style given; a port named “Scoreboard14” would be valid, as long as the remote machines referring to it use that name. Note that port names must not contain spaces or characters other than A..Z, a..z and 0..9.

Tracing

To aid in debugging and resolution of network problems, Remote Serial Server includes an activity trace which shows all communication between the server and its clients. For any “Data in” or “Data out” transfer, holding the mouse over the command will show a hex and ASCII detail of the data moved.

