Totaal Software

StageRace

StageRace Message Protocol

Subjected version: 2000.1.12– 2000.1.20

Date: November 2001

Window messages 2
Message format 2
Message extensions (atoms) 3
Message definitions 3
Technical references 6

This document briefly summarises the message format as well as the specific messages that are defined for the communication between StageRace and other software entities.

Window messages

Window messages are used within Windows® to communicate between separate application windows.

Window messages are identified by their unique message identifier. Message identifiers are numerical values, but are kept hidden by string identifiers, such as “WM_USER”.

On start-up of StageRace, all appropriate message strings are registered with Windows®.

Other applications can do the same, arranging all the applications to have thes same correct and unique numerical identifiers to match the string identifiers.

Message format

The Windows® API function for inter-window messaging are defined like the one below:

```
BOOL PostMessage(
  HWND hWnd,        // handle of destination window
  UINT Msg,         // message to post
  WPARAM wParam,    // first message parameter
  LPARAM lParam     // second message parameter
);
```

Ergo the message format is like this:

Destination handle	Message ID	First parameter	Second parameter
HWND	UINT	WPARAM	LPARAM
32-bits window handle	*32-bits unsigned integer*	*32-bits user-defined bit-string*	*32-bits user-defined bit-string*

We suggest messages to be posted as global broadcast. In this case*Destination handle* = HWND_BROADCAST. This will cause your message to be sent to all top-level windows, but only the appropriate window will respond anyway due to the specific *Message ID*.

Message extensions (atoms)

Applications can register strings with Windows in exchange for an identifier of the ATOM type. These atom identifiers are used by the messages sent across windows to pass strings “by pointer”.

Strictly speaking, someone is responsible to clean up atom strings after use. In the following definitions StageRace will take responsibility to clean up the atoms.

Message definitions

Note that numerous parameters are divided in two 16-bits parts. We recommend to set values that are indicated as ‘not used’ to be set to NULL.

Client to StageRace

To arrival dialog

1. WM_STAGERACE_AR_SETCOMP

- First parameter (W): upper bytes: rank [unsigned short int]
- First parameter (W): lower bytes: start no. [unsigned short int]
- Second parameter (L): time [ms] [unsigned int]

Request to the arrival dialog to submit the competitor in the arrival list

Note 1: For time trials: the time is the gross time
Note 2: For time trials: the rank is ignored
Note 3: Illegal competitors are ignored without further notice

Return message: *none*

2. WM_STAGERACE_AR_DECLASSCOMP

- First parameter (W): upper bytes: *not used*
- First parameter (W): lower bytes: StaNo [unsigned short int]
- Second parameter (L): *not used*

Request to the arrival dialog to declass the competitor

Note: Illegal competitors are ignored without further notice

Return message: *none*

3. WM_STAGERACE_AR_GETCOMPFROMSTANO

- First parameter (W): upper bytes:*not used*
- First parameter (W): lower bytes: start no. [unsigned short int]
- Second parameter (L): requesting window [HWND]

Request to the arrival dialog to send this competitor's rank and race time

Return message: WM_STAGERACE_AR_INFOCOMP

4. WM_STAGERACE_AR_GETCOMPFROMTIME

- First parameter (W): time [ms] [unsigned int]
- Second parameter (L): requesting window [HWND]

Request to the arrival dialog to send rank and race time of the first competitor with a better time than provided

Note: For time trials: the time is the net time

Return message: WM_STAGERACE_AR_INFOCOMP

5. WM_STAGERACE_AR_GETCOMPFROMRANK

- First parameter (W): upper bytes: rank [unsigned short int]
- First parameter (W): lower bytes:*not used*
- Second parameter (L): requesting window [HWND]

Request to the arrival dialog to send start no. and race time of the competitor with the rank as provided

Return message: WM_STAGERACE_AR_INFOCOMP

6. WM_STAGERACE_TT_GETSTARTTIME

- First parameter (W): upper bytes:*not used*
- First parameter (W): lower bytes: start no. [unsigned short int]
- Second parameter (L): requesting window [HWND]

Request to the time trial arrival dialog to send the start time of a competitor

Return message: WM_STAGERACE_TT_INFOCOMP

To StageRace view

WM_STAGERACE_GETCOMPFROMSTANO

- First parameter (W): upper bytes:*not used*
- First parameter (W): lower bytes: start no. [unsigned short int]
- Second parameter (L): requesting window [HWND]

Request to the document to send this competitor's name, team and team code

Return message: WM_STAGERACE_COMP

StageRace to client

From arrival dialog

1. WM_STAGERACE_AR_INFOCOMP
(sent to the requesting window)

- First parameter (W): upper bytes: rank [unsigned short int]
- First parameter (W): lower bytes: start no. [unsigned short int]
- Second parameter (L): time [ms] [unsigned int]

Sends the arrival information of a particular competitor in the arrival list

Note 1: In case the requested competitor/rank/time is not available, start no. will be set to –1 while other fields do not necessarily consist of any valid information
Note 2: For time trials: the time is the net time

Return message: *none*

2. WM_STAGERACE_TT_INFOCOMP
(sent to the requesting window)

- First parameter (W): upper bytes: NULL
- First parameter (W): lower bytes: start no. [unsigned short int]
- Second parameter (L): time [ms] [unsigned int]

Sends the starting time of a particular competitor in the available/arrival list

Note: In case the requested competitor/rank/time is not available, start no. will be set to –1 while other fields do not necessarily consist of any valid information

Return message: *none*

From StageRace view

WM_STAGERACE_COMP

- First parameter (W): upper bytes: name [atom identifier]
- First parameter (W): lower bytes: start no. [unsigned short int]
- Second parameter (L): upper bytes: team [atom identifier]
- Second parameter (L): lower bytes: team code [atom identifier]

Sends the global information of a particular competitor in the document

Note 1: Text strings are provided as atoms. Receiving window is responsible for cleaning up atoms
Note 2: In case the requested competitor is not available, start no. will be set to –1 while no atoms being created thus the other fields not necessarily consisting of any valid information
Note 3: Refer to *Message extensions (atoms)* for information on the atoms that are used for this message

Return message: *none*

Non-StageRace

For communication between the Time Trial Controller and the Score Board Controller from Totaal Software, the following two additional messages are in order.

1. WM_TTC_GETTIME

- First parameter (W): NULL
- Second parameter (L): requesting window [HWND]

Request to the Time Trial Controller to send the current time in the race

Return message: WM_TTC_TIME

2. WM_TTC_TIME

- First parameter (W): upper bytes: NULL
- First parameter (W): lower bytes: start no. [unsigned short int]
- Second parameter (L): time [ms] [unsigned int]

Sends the arrival time and possible start no. of an arrived competitor from the Time Trial Controller to the Score Board Controller

Note: Start no. may be zero in which case the message contains the current time

Return message: *none*

Technical references

The inter-window communication is discussed in the library of the Microsoft Developer Network (MSDN).

- About Messages and Message Queues
  http://msdn.microsoft.com/library/en-us/winui/messques_3pgz.asp
  Start page for window messaging subjects

- PostMessage
  http://msdn.microsoft.com/library/en-us/winui/messques_87mt.asp
  Documentation on the PostMessage SDK function for posting messages

- RegisterWindowMessage
  http://msdn.microsoft.com/library/en-us/winui/messques_10h1.asp
  Documentation on the RegisterWindowMessage SDK function for registering a message in Windows®

- Atom Functions
  http://msdn.microsoft.com/library/en-us/helplib/atoms_1z3n.asp
  Start page for atom functions, with shortcuts to GlobalAddAtom, GlobalGetAtomName and GlobalDeleteAtom