SALIERI Language Tutorial

Update Information from Version 2.0 to 2.0.7c

Revision 1.0, 97/10/23 (by Holger H. Hoos)

This document contains a brief overview of the most significant changes between the SALIERI Language, Version 2.0 and the current Versions 2.0.7c; it is intended as an update of the SALIERI v.2.0 Tutorial and refers to the respective sections. For more information on the SALIERI System and Language, the latest releases of the SALIERI System, and the documentation, please visit the SALIERI System and Language Homepage at `www.informatik.tu-darmstadt.de/AFS/CM/SAL`.

New SALIERI for Windows graphical user interface

The current version of the SALIERI System for Windows provides a interactive development environment containing Workspace Browsers, Program Editors, Note Viewers and an Online Help System (the latter is currently only available in German, but english online documentation is provided in form of the HTML-based SALIERI Language Online Reference and Help).

The functionality of the old command-line interface is still available in form of the System Dialog window. Use the Program Editor to create and edit SALIERI programs; program text can be highlighted using the mouse (hold left button) or cursor keys (hold shift key); highligted text can be executed using Ctrl-E or Edit/Exec from the Menu.

The SALIERI for Windows integrated development environment is quite self-explanatory, in addition, extensive documentation will be soon available.

(updates Introduction, Sections 2.1, 2.2)

Object names

Object are no longer allowed to contain dots (".") as these will be used for other purposes in future versions of SALIERI. You can use underscores ("_") instead of dots to structure object names.

(updates Section 3)

Integer range

Integers have now an extended range from -2^{31} to $2^{31} - 1$. Consequently, the theoretical length limits for character strings, note sequences, and lists are $2^{31} - 1$. In practice, however, the maximal lengths are determined by the available memory and performance considerations.

(updates Section 4.2, 4.4, 5.1, 8.1)

Secondary parameters (M,I,T) for sequences

The current notation for the tempo (M), intensity (I) and instrument (T) parameters in sequence constants will soon become obsolete when SALIERI supports the full GUIDO Notation Format. Try to avoid specifying these parameters directly in sequence constants and use the functions *accel, cresc,* and *instr* instead.

(updates Section 5.1)

Syntax checking and error messages

As the SALIERI Language is now compiled in to an internal intermediate code format to improve performance, syntax checking for function objects is done when a function object is defined (whereas formerly it was only done when executing a function).

(updates Section 6.2, 11)

Blocks and Macros

Blocks are no longer semantically equivalent to parameter-less macros; now, blocks are executed in their declaration context, i.e., in the context of the function body in which the respective block constant was defined. This becomes only significant when blocks are passed as function parameters and executed in a context different from their declaration context. Most existing SALIERI programs will not be effected by this modification; but now, it is possible to define new control structures in a more consistent and robust way.

(updates Section 7.1)

Function *play* replaces *midiPlay*

The internal function *midiPlay* has been renamed to *play*. To run old SALIERI programs using this function, just define "midiPlay := play" on the command-line or in your program file.

(updates Section 9)

New function *view*

A new internal function *view(segm)* can be used to display sequences or segments *segm* in a Note Viewer window using conventional music notation. The Note Viewer module is a pre-release version and might cause trouble when trying to display very complex musical material.

view(segm,str) can be used to additionally specify *str* as the name or title of the musical fragment being displayed.

(updates Section 9)

New function *writeStr*

A new internal function *writeStr(any1,...)* can be used to write arbitrary objects to the System Dialog; different from *write*, the output is not automatically succeeded by a newline character.

(updates Section 9)

Function *execDOS* **disabled**

The internal function *execDOS* does not work in the current version of SALIERI. Probably it will be removed from future versions of SALIERI.

(updates Section 9)