

ConsoleTools Lua Framework

version 1.0 | 05/23

provided by notenlektorat.de

Overview

The notenlektorat ConsoleTools Lua framework provides Dorico users with the means to organise arbitrary Lua scripts as “tools” and access them by way of keycommand-controlled “tool slots”.

In the context of the framework, the term tool can mean the execution of:

- an arbitrary Lua file (e.g., a recorded macro)
- a function from a table that was returned by a Lua file (i.e., an external tool library)
- a Dorico command string
- a local Lua function declared during setup, or a global Lua function

It is possible to pass arguments to tools. Default arguments can be specified for any tool during setup, and tools can be called with explicit arguments via the console, overriding any such default. Tools can also be bound to variables in the Lua environment, which can be called from the console.

A tool slot may hold multiple tools, with functions for cycling and toggling between them provided for use via keycommands. Assignment of tools to slots can be changed via console, or by application of custom profiles; cycling and toggling between profiles is likewise possible with keycommands.

The framework also provides functionality to store and recall basic documentation for tools.

Availability

The ConsoleTools framework is available as a full version (for purchase) and a free limited version. The versions differ as follows:

	Full Version	Limited Version
Number of tools that can be set up:	unlimited	no more than 10
Number of tool slots that can be set up:	unlimited	no more than 2
Number of variables that can be bound to tools:	unlimited	no more than 3
Default arguments can be set when allocating tools:	yes	no more than 20 times per session
Immediate tool execution of a tool cycled/toggled to:	yes	for no more than 3 tools at once
Tool source file reloading, to reflect <i>ad hoc</i> code edits:	yes	no
Included auxiliary library can be upgraded:	yes	no, only extendable

CAVEAT UTILITOR

- Setting up notenlektorat ConsoleTools is not particularly complicated. It is not an entirely straight-forward process, either, as it involves manual editing of Dorico’s JSON data for keycommands. If in doubt, make backups of any relevant files before editing.
- The framework does **not** contain a noteworthy amount of Lua scripts for use with Dorico, merely some arbitrary examples to demonstrate how to set up and use a tool collection.
- No programming skills are required for using ConsoleTools. However, a basic understanding of the Lua language will be beneficial when using some of the advanced features.
- The framework makes use of Lua’s `loadfile` function (in “text chunks only” mode), which by its nature comes with potential security issues. Users are advised to only run trusted scripts with the framework, just as they would when using Lua scripts directly in Dorico.