



Scilab 6.0 - Release Notes

February 15th 2017

32/64-bit Linux, Mac OS X, 32/64-bit Windows

This document details new features, changes and improvements found in Scilab 6.0.0 release.
Scilab 6.0.0 is available for download at: <http://www.scilab.org/download>

Contents

What's new in Scilab 6.0	2
New language parser & new core	2
Utilities for code productivity.....	2
Language changes	3
Data Structures	3
API modification	3
Xcos.....	3
ATOMS.....	4
Newsfeed	4
Help pages.....	4
Licensing changes.....	4
Feature changes and additions.....	5
Obsolete functions or features	5
Platform-specific limitations.....	5
Complete changelog	6

Scilab 6.0 is a major release of Scilab, the open source platform for numerical computation.

Scilab is an open source software for modeling, simulation and scientific data analytics. Its development started in 1994 in the french research, and is now supported by the company Scilab Enterprises. With the evolutions of the technologies in numerical computation, the software witnessed several deep changes, and Scilab 6 is one of them. Structural modifications at the core make it now suitable for future functional evolutions. In our development we emphasized on the importance of backward compatibility with Scilab 5.x.x. However if you find bugs or incompatibilities, please report them on our Bug Tracker: this will help get a better product, faster, which ultimately will be benefit you too.

What's new in Scilab 6.0

In summary, the main new features are:

- ▶ New language parser and interpreter and new core ensuring:
 - Support for bigger data sets, thanks to dynamic memory usage. No need for `stacksize` anymore.
 - Better performance for some objects (cells and structs).
 - Clearer, less ambiguous, language syntax.
 - Xcos also uses the new re-written Scilab engine.
- ▶ New code productivity features: full-featured debugger, profiler / coverage tool, and "lint"-like commands.
- ▶ Newsfeed, providing a continuous feed of news, tips, and general communication from the community and from the Scilab Team.
- ▶ Licensing change: Scilab is now released under the terms of the GNU General Public License (GPL) v2.0
It is still also available under the terms of the CeCILL v2.1.

New language parser & new core

This release includes a major rewrite of Scilab's internal core, for better usability and performance: the changes to the visible aspects of the software are limited.

The first big internal change, is a complete new language parser and interpreter, with the following changes:

- ▶ The language is much more coherent and predictable,
- ▶ The grammar is stricter, but more consistent,
- ▶ Error handling has been improved (but may still be incomplete: please report issues you find),
- ▶ The language has slightly better compatibility with Matlab®'s language (for example control instructions such as switch/case statements).

The language changes are not all backward compatible. You may have to update your scripts/programs/modules, but it will be very simple in pretty much all cases. In particular, the following changes may require modifications to existing code:

- ▶ Literal strings cannot mix single-quote and double-quote delimiters: they have to start and finish with either both single-quotes, or both double-quotes.
- ▶ Usage of commas (to put multiple instructions on the same line) has been clarified: it can not be used following control instructions such as try.
- ▶ The operation plus and minus with the empty matrix will now yield an empty matrix. To simplify the transition of your code base, a specific warning is issued when such operation is detected.
- ▶ To determine the exact line of code generating a warning, an option warning("stop") has been added.
- ▶ A retrocompatibility flag for empty behaviour operations is also available. It will be deprecated in a future scilab release. See oldEmptyBehaviour.

The second big internal change, is the ability to manipulate bigger data sets, without exhausting system resources. This includes in particular:

- ▶ A much better memory management. Memory usage is now only limited by the machine's memory. This was a much-needed, and much-requested feature. There is no need anymore for stacksize.
- ▶ Support for bigger data sizes, in particular with better support for HDF5 files.
- ▶ Some new and improved data types: 64-bit integers; ability to define user types in C++ modules. Also, matrices with more than two dimensions are now implemented natively.
- ▶ Native data structures (cell, struct), as well as multidimensional matrices, which yield, in most cases, better performance, both in memory usage, and execution speed. There could be some performance degradation in some cases: please feel free to report them.

Utilities for code productivity

This release includes also several features to write better and more efficient Scilab scripts:

▶ A full-featured debugger is now included. It can run Scilab scripts or functions step-by-step, break on (conditional) breakpoints, in particular only when an exception is set... Run "help debug" for more information. However, The debugger does not have a graphical user interface, and is not integrated with the editor: it is only accessible from the command line. As for all features, we are open to third-party contributions (in code or otherwise) to the development of a GUI-based debug.

▶ A new profiler and coverage tool has been added. It replaces the removed "profile" function. It produces HTML report for the time spent (profile) on each line. The documentation is lacking at this stage. Also, it is known not to work on Xcos, and probably does not work well on complex code. Example of use:

```
covStart(isempty)
// starts profiling the "isempty" macro;
// profiling a whole library can be done
// with:covStart("elementary_functions")
for i=1:1e5; isempty(i); end

// execute some code - this will be profiled
covWrite("html", "TMPDIR/coverage")

// writes the profiling/coverage report as
// "report.html" in "TMPDIR/coverage" folder
covStop();

// stop profiling
```

▶ A "lint"-like command, slint, has been added, for code static analysis: it detects unclear, risky or suspicious code. It is not documented yet, however. Usage:

```
slint("<sci_filename>")
```

Language changes

Some small changes have been applied to the language syntax, aiming at more clarity and less ambiguity. Some changes are not compatible with 5.5; code written for Scilab 5.x might need to be modified in order to work in Scilab 6.0.

- ▶ Declaration of a number on two lines is no longer allowed: 1.. \n 2.
- ▶ Comment blocks on multiple lines /* ... \n ... \n ... */ are now possible.
- ▶ 1./M is now parsed as 1 ./ M instead of 1 / M.
- ▶ Declaring strings using non-homogenous delimiters ("string' or 'string") is no longer allowed.
- ▶ (a=b) executed as a == b is now deprecated and returns an error.
- ▶ Function definitions can finish with end instead of endfunction.
- ▶ {} and [] are now distinct operators. Matrices can no longer be defined using {} operators. These are reserved for cell definition.
- ▶ Shortcut and element-wise boolean operators are now distinct. && and || are new shortcut boolean AND and OR, while & and | are element-wise operations and will not shortcut one of the operand. Both shortcut and element-wise operators are evaluated from left to right.
- ▶ Syntax %i:10 is now deprecated (only real scalars can be used).
- ▶ while ... else ... end control instruction is not supported anymore.
- ▶ Transposed matrix can now use the extraction operator

directly A'(1, 2)

▶ Function without output argument cannot be called in assignation expression anymore:

```
function foo()
    //...
endfunction
val = foo() // now returns an error
```

Data Structures

cells and structs are now native types, hence improving performances.

cells:

- ▶ insertion and extraction must be done via () or {}.
- ▶ .dims and .entries fields have been removed, please use size and () instead.

struct:

- ▶ .dims field has been removed, please use size instead.

hypermatrix:

- ▶ hypermatrices are natively managed (without mlist overloading).
- ▶ typeof function now returns real type like constant, string, ... instead of hypermat
- ▶ type function returns real type like 1, 10, ... instead of 17 (mlist).
- ▶ .dims and .entries fields have been removed, please use size and () instead.

API modification

A new set of C APIs to write C or C++ extensions (toolboxes) to Scilab. It allows defining native functions (commonly called "gateways"), getting input parameters for such functions, setting return parameters, accessing local variables, using common helper functions for accessing environment information (such as warning level), generate errors...

It also includes ways to overload existing Scilab functions to support additional parameter types (see help scilab_overload). Finally, you can call back Scilab functions (macros and built-in functions) from your gateway (see help scilab_call).

User-defined functions written in C or C++ (gateways) must now use a void* pvApiCtx name as a second parameter instead of any unsigned long l. This is now required for some macros, such as Rhs, to work.

For example: use int sci_Levkov(char *fname, void* pvApiCtx) instead of int sci_Levkov(char *fname) or int sci_Levkov(char *fname, unsigned long l).

Xcos

Xcos, the dynamic system modeler and simulator packaged with Scilab saw some major improvements coming with this new version. It now benefits from the new engine of Scilab 6. The major rewrite of the data structures should enable huge models to load and save faster. The memory usage on diagram edition is also slightly reduced.

Xcos also uses the new re-written Scilab engine, and is now running in native code, which should provide much better performance as well as support for much bigger models. Xcos can load and run models created with earlier versions of Scilab, but models saved with Xcos 6 cannot be loaded in earlier versions.

On top of this significant improvement of the behaviour of Xcos, two new features appear in this version for ergonomiy enhancement

- ▶ Added a new link style (Optimal) for automatically finding the optimal route.
- ▶ Automatically reposition split blocks for better-looking layout.

Concerning the compatibility of diagrams written in Xcos 6 to earlier versions::

- ▶ ZCOS and XCOS file formats have evolved to reduce the duplicated information. Scilab 5.5.2 is able to open the newly saved files, but the ports have to be repositioned manually.

Minor modifications have been introduced in two blocks:

- ▶ INVBLK: add a divide by zero parameter to ignore the error
- ▶ PRODUCT: add a divide by zero parameter to ignore the error

And a solver...

- ▶ Implicit fixed-size step ODE solver added: Crank-Nicolson 2(3). Added to the CVode package, it also benefits from the CVode rootfinding feature.

One more thing: The palette browser has been improved! The following features were included:

- ▶ Search engine
- ▶ History (go forward or backward)
- ▶ Drag and drop multiple blocks at once
- ▶ Navigate using the keyboard arrows
- ▶ (a=b) executed as a == b is now deprecated and returns an error.
- ▶ Add blocks to the most recent diagram by typing ENTER
- ▶ Dynamic palette with the last used blocks
- ▶ Zoom using CTRL(+), CTRL(-) and CTRL(mousewheel)
- ▶ Load SVG icons

ATOMS

- ▶ A ``tbx_make()` function is added to build a toolbox following the toolbox directory structure convention
- ▶ ATOMS builder functions are now less dependent on the script files in the `macros`, `help`, `etc`, `src` and `sci_gateway` directories. These functions will do nothing (but warn the user) if they find no target to build:
 - ``tbx_builder_macros``: compiles the ``.sci`` files present in `macros/`` directory into the toolbox library. If `buildmacros.sce`` or `builder.sce`` script in `macros/`` are present, executes them instead;
 - ``tbx_builder_help``: compiles the help for each language (`la_`` LA) directories in `help/``. If `builder_help`` script is present in `help`, executes it instead;
 - ``tbx_build_loader``: builds a default loader that mimics the skeleton ``.start`` files. If a ``.start`` file is found in `etc/`` builds a loader calling this script;
 - ``tbx_builder_src``: scans the subdirectories under `src`` for builder files and executes them
 - ``tbx_builder_gateway``: scans the subdirectories under `sci_gateway`` for builder files and executes them

Newsfeed

New in Scilab 6.0.0. A new view is available in the visual environment of Scilab: the "Newsfeed". It provides a continuous feed of news, tips, and general communication from the community and from Scilab Enterprises. This is technical and/or professional information, about usage of Scilab, help for specific features, usage examples... We may include some professional information from Scilab Enterprises, only information directly useful to the user. We see it as way to create an open line of communication with (and between) the users.

Help pages

The following pages have been fixed or improved:

- ▶ members, part, ode, ode_optional_output, ode_roots, plot2d, roots, printf, sprintf, iconvert, stdev, xlabel, and_op, or_op, permute, tree2code, %helps, scilab|scilex, flipdim

These pages were rewritten:

- ▶ consolebox, double, isoview, pixel_drawing_mode, householder, or, and, format, typeof, brackets, setlanguage, sleep, isinf, bitor, bitxor, bitand, macr2tree, geommean, clf

An additional effort was set on reorganizing parts of the help:

- ▶ else, elseif, end, try, sciargs, global, halt, empty, power
- ▶ pixel_drawing_mode, show_window, twinkle, uigetcolor, winsid, xdel, xgrid, xname, xnumb
- ▶ repmat, sign, nthroot, lstsize, cell2mat, cellstr, ind2sub, sub2ind, and, or, unwrap

Control system (CACSD) and Signal Processing help pages have been sorted out.

In the category Signal processing, a new Convolution - correlation subsection was created. The following functions were sorted out in existing subsections: `wfir_gui`, `filt_sinc`, `hilb`, `fft2`, `fftshift`, `ifftshift`, `hilbert`, `cepstrum`, `conv`, `conv2`, `convol2d`, `xcor`, `corr`, `hank`, `mrfit`, `frfir`

Colormaps and GUI/Menus subsections were created.

Some translations were added:

- ▶ In Russian: a homepage has been added
- ▶ In French: `format`, `typeof`, `isoview`, `ndgrid`, `bench_run`, `consolebox`, `harmean`, `sleep`, `strtod`, `permute`

Licensing changes

New in Scilab 6.0.0. Scilab is now released under the terms of the GNU General Public License (GPL) v2.0, which you can find in the COPYING file at the root folder of this distribution. Prior to this version, Scilab was licensed under the terms of the CeCILL v2.1, and continues to be available under such terms, which you can find in the same COPYING file. This change has been made pursuant to article 3.5.4 of the CeCILL v2.1.

In this release, the Intel Math Kernel Library and the FFTW library optimized by Intel Math Kernel Library are included in Scilab. They are subject to the Intel EULA.

Feature changes and additions

- ▶ Addition or subtraction with an empty matrix now returns an empty matrix. `scatter/scatter3` plot with different mark colors is now available.
- ▶ `parulacolormap` is now available.
- ▶ `name2rgb` can now handle a single string and a matrix of strings.
- ▶ `isoview`, `isoview on`, `isoview off`, `isoview(idGraphics, "on"|"off")` are now supported.
- ▶ `twinkle` and `twinkle(n)` are now supported: by default, the current element `gce` blinks.
- ▶ `replot` has been upgraded:
 - a bound set to `%inf` now gets the position of the most marginal object,
 - `replot()` can now be used to reframe axes to all their contents,
 - option `"tigh_limits"` added,
 - Any handle having some Axes as direct children -- as `uicontrol-frame` -- is now supported.
- ▶ `householder` can now return the reflection matrix, and has a demo.
- ▶ `ndgrid` can now works with any types of homogeneous data
- ▶ `permute` now supports arrays of rationals.
- ▶ `bench_run` can now return its results and/or record them in a file
- ▶ `typeof(.., "overload")` allows now to get the overloading type-code of an object
- ▶ `sign` can now handle a sparse matrix.
- ▶ `sleep(..,'s')` allows now to specify the duration in seconds.
- ▶ `real`, `imag`, `conj` and `isreal` now accept rational fractions.
- ▶ A call stack limit has been introduced. Default maximum depth is setup to 1000 and can be changed by `recursionlimit` or through the Preferences interface.
- ▶ The floating point exception mode `ieee` is now set to 2 by default: floating point exceptions now produce `Inf` or `Nan`, and do not cause any error. The previous behavior can be recalled by simply calling: `ieee(0)`.
- ▶ The graphics entity "Datatip" has its property `z_component = 'on|off'` changed to `display_components = 'xyz'`; it is now possible to choose which components to display, and in which order. Loading old `*.scg` files containing datatips with `"z_component"` property may not display the z component.
- ▶ Valgrind error detection added to `test_run` (on Linux only).
- ▶ `amell` now:
 - checks if its parameters are real numbers,
 - throws an error if the second parameter is not a scalar.
- ▶ The use of I/O console is now allowed with the following functions: `mget`, `mgetl`, `mgetstr`, `mput`, `mputl` and `mputstr`.
- ▶ `mclearerr` now returns a flag indicating the file identifier validity.
- ▶ `fileinfo` can now take a row vector as input.
- ▶ `msprintf` does not return an error message anymore when there are too many input arguments (more values that format needs).
- ▶ `deletefile` can delete multiple files at once.
- ▶ `exec` of macro executes the body in the current scope, but the prototype must have zero input and output arguments.
- ▶ `error`: an error number in input is deprecated.
- ▶ `impl`: Recall `impl` with the same parameters as in its previous stop is now available.
- ▶ `ode`: `y0` is restricted to a column vector.
- ▶ `pppdiv`: Return a scalar of type 'constant' when the rank is 0.
- ▶ `pdiv`: Return a matrix of type 'constant' when all the rank are 0.
- ▶ `test_run` can now take `"[]"` as argument to be used on console; for instance: `test_run string [] no_check_ref`.
- ▶ `typeof(:)` and `typeof(n:$)` now return "implicitlist" instead of respectively "constant" and "size implicit".

- ▶ `linspace(a, b, n<=0)` now returns `[]` instead of `b`.
- ▶ `strange([])` now returns `%nan` instead of `[]`, as all other functions for statistical dispersion.
- ▶ `stdev(x, dir>ndims(x))` now yields an error instead of returning `zeros(x)`.
- ▶ `write`: Writing string or string matrix in a file does not add blank space before each value.
- ▶ `bitor`, `bitxor` and `bitand` are upgraded:
 - positive signed encoded integers are now accepted.
 - inputs with new `int64` or `uint64` encodings are now accepted.
 - operands with mixed types or/and `inttypes` are now accepted.
 - distributive input scalars as in `bit###(scalar, array)` or `bit###(array, scalar)` are now accepted.
 - results with decimal-encoded integers $> 2^{32}$ are now correct.
 - decimal-encoded integers $> 2^{52}$ are now supported up to the biggest `1.80D+308`.
 - `bitxor` is now vectorized and fast.
- ▶ The zoom rubber box now can start/finish from points lying outside the axes bounds.
- ▶ The zoom rubber box now can select multiple and overlying axes at once. Changed behavior of scroll zoom, scrolling over overlying axes will zoom all of them together, using the CTRL key while scrolling will zoom all axes in the current figure.
- ▶ The graphics entity "Datatip" has a new property `detached_position` which accepts `[]` or a 3-components vector to set the position in axes coordinates to draw the datatip text box.
- ▶ `MPI_Create_comm` create a new communicator from `MPI_COMM_WORLD` using MPI world ranks.
- ▶ The grand non-free `fsultra` generator was removed.
- ▶ The original `rpoly` algorithm was removed in favor of a C++11 implementation
- ▶ When the view property of Axes object is set at 2d, the rotation becomes impossible.
- ▶ The zero-pole-gain (`zpk`) representation added for linear dynamical systems.
- ▶ It is now possible to add a title to the axes via the "Label -> Title" context menu entry

Obsolete functions or features

- ▶ `maxfiles` is now obsolete.
- ▶ `isoview(xmin,xmax,ymin,ymax)` is deprecated. Please use `isoview("on")`, `replot(..)` instead.
- ▶ `eval3d` will be removed from Scilab 6.1. Please use `ndgrid` instead.
- ▶ `strcmpi` is deprecated. Please use `strcmp(..,"i")` instead.
- ▶ `square` will be removed from Scilab 6.1. Please use `gcf().axes_size` and `replot` instead.

Platform-specific limitations

All functionalities of Scilab are present on all platforms, with a few exceptions:

- ▶ TCL/TK functionalities, as well as functionalities which depend on them (as `ged` or `object_editor`), are not available on Mac OS.
- ▶ Some functions using specific Windows platform are only available on Windows
- ▶ Compilation of C/C++ or other native code in additional modules (toolboxes), and interface with additional native third-party libraries may depend on the platforms, and may not always be supported on all platforms.

Complete changelog

For a precise list of changes (in particular, bug fixes) between the 5.5.2 release and the 6.0.0 release, please see the CHANGES file at the root of the installation.

Scilab is a trademark registered by INRIA with exclusive use by Scilab Enterprises SAS.

Matlab® is a trademark registered by the Mathworks.