

List of changes in the program GraphiXT

Below is the list of changes for each version of the file GraphiXT.exe that was available for download from www.graphixt.com , up to the current version (1.21.4.4).

Explanation of the version number format:

GraphiXT:	AA.BB.F.P
Function file:	XX.YY.ZZ.F
Parameter editor:	XX.YY.VV.P

The first two numbers (AA.BB) are the main components of the GraphiXT version number. The numbers AA and BB only increase after improvements in the GraphiXT user interface, addition of new features, bug fixes, or revising the documentation. The other two components (F and P) indicate compatibility of the program GraphiXT.exe with the function file and the parameter editor, respectively. Thus, F and P can be called the “program interface version numbers” of the function file and the parameter editor, respectively. Those two numbers only change after a change in the rules of communication between GraphiXT.exe and the other two programs. In order to be able to load the function file and the parameter editor, the number F must coincide with the fourth component of the function file version number, and the number P must coincide with the fourth component of the parameter editor version number.

The first two components of the function file and parameter editor version numbers (XX.YY) indicate the set of parameters and functions that define the simulation model. Thus, XX.YY can be called the “model version number”. The numbers XX and YY only increase after adding or removing model parameters and functions or after changing the meanings of model parameters and functions. Those two numbers must be the same both for the function file and the parameter editor, because both those programs must correspond to the same set of model parameters and functions. The third component of the function file version number (ZZ) increases after improving the simulation algorithm or fixing simulation bugs. The third component of the parameter editor version number (VV) increases after improving the parameter editor user interface or fixing its bugs or revising the documentation of the plug-in that solves the kinetic equations.

In the following list of changes, the differences between the indicated GraphiXT version and the previous version are listed. Note: the changes given for the first entry (v.1.10.4.3) are the differences from the first English version (v.1.00.4.2), which was released in February, 2012.

1.10.4.3 (2012-07-06):

1. The nonlinear fitting dialog window has been improved. Now, the names of parameters, functions and datasets, as well as parameter values and limits of fitting ranges are shown in respective grid controls, whose format is similar to that of Microsoft Excel worksheets (in the previous versions of GraphiXT, linked list boxes were used for that purpose). In addition, an option to mark each fitted dataset as “used” or “unused” has been added.
2. In the current version, it is no longer required to check at least one of the two checkboxes “Connect points” and “Show points as symbols” in the curve format dialog window. If neither checkbox is checked, then the corresponding curve becomes “hidden”, i.e., it is not plotted in the graph window, its name is not shown in the legend and its data are not taken into account when rescaling the axes. However, it is still possible to analyze the data of hidden curves (e.g., to use them for nonlinear fitting, smoothing, etc), or to use them in formulas. If a curve is hidden, then its options can only be accessed by opening the corresponding list of curves (menu commands “Graph options / Free curves...”, “Graph options / Select...”, “Simulation options / Formulas...”).

3. A new menu command “Graph options / Free curves...” has been added. By selecting that command, a dialog window with a list of all free curves (including the “hidden” ones) is opened. In that list, it is possible to edit options of any free curve.
4. Two commands “Copy formula” and “Cut formula” have been added to the formula context menu. The difference of those commands from the commands “Copy curve” and “Cut curve” is that the formula text and the values of local parameters are copied to the clipboard instead of the point coordinates. Accordingly, a command “Paste formula” has been added to the graph context menu.
5. The option “Independent set of argument values” of the formula dialog window has been made more flexible: if that checkbox is checked after checking any one of the other two checkboxes (“X axis range” or an existing dataset), then, after clicking “OK” or “Apply”, the point abscissas corresponding to the previous option will be used (i.e., they will not be recalculated using the entered limits and the number of points). The abscissas are only recalculated after changing a limit value or the number of points.
6. The build dates of the program GraphiXT and the loaded simulator plug-in are shown in the dialog window “About GraphiXT”.
7. Several minor bugs have been fixed, many minor improvements of the user interface have been made.

1.11.4.4 (2012-07-29):

1. Sets of abscissa values of free curves and formulas (the “X sets”) are now treated as a separate category of datasets (in the previous versions, each set of abscissas was treated as a part of a particular free curve or formula data). Any number of free curves or formulas can be linked to an X set. When an abscissa of a point is changed, the corresponding abscissas of all other curves linked to the same X set are changed, too. When two or more free curves are pasted or imported from a text file, they are initially linked to the same X set. Linked curves can be unlinked using the checkboxes “Unlink this curve” or “Unlink all curves” that have been added to the curve format dialog window. In addition, it is now possible to specify that abscissas of skipped points must be the same for all curves linked to a particular X set (this is done using a corresponding checkbox of the curve format dialog window). Properties of X sets can also be changed using a new menu command “Graph options / X datasets...”.
2. In the current version, if a dialog window with a list of model functions or with their properties is open in GraphiXT, then major modifications of model parameter values done in the parameter editor are not accepted until the user closes that dialog.
3. The GraphiXT user’s manual has been updated to reflect the recent changes in the program.

1.15.4.4 (2012-09-08):

1. The current version of GraphiXT can display function data in table format. Each data table is associated with a particular graph window, and there is only one data table for each graph window. Two table modes are possible: in “partial” mode, the table only shows the data corresponding to the points that are visible in the graph window (this is the default mode); and in “complete” mode, all the data associated with the given graph window are shown in the table. In a table window, function values can be edited either directly (by entering values into corresponding cells), or by copying and pasting. Switching between the table window and the corresponding graph window or between the two modes of the data table can be done using the corresponding menu commands or toolbar buttons.
2. A new command “Create free curves” has been added to the “Graph options” menu. This command is used to create one or more free curves by specifying the range of their x and y values.

3. An option to show the current time edit control next to the time slider, when the active window is a time graph window, has been added.
4. Fixed a crash when attempting to delete a formula linked to model time or coordinate values when there are other formulas linked to independent X datasets in the same graph.
5. Several other minor bugs have been fixed, several minor improvements of the user interface have been made.
6. The GraphiXT user's manual has been updated to reflect the recent changes in the program

1.19.4.4 (2013-06-21):

Most of the changes in comparison with version 1.15 are related to extending the computational programming capabilities of GraphiXT:

1. A new menu "Programming" has been added, which groups the menu commands providing access to programming objects (i.e., formulas, subroutines, arrays, parameters and DLL functions) and to their properties.
2. Formula editor and the text editor (the "Info" window) have been re-written on the basis of the code "Crystal Edit" by Andrei Stcherbatchenko. All editing history of each program and of the text in the "Info" window is preserved even after closing the editor window, so that all editing actions can be undone or redone when that text is opened again.
3. Several new built-in functions have been added, including the "loc(...)" function, which is used to pass addresses of variables, arrays and functions to other functions (also see below).
4. Support for user-defined subroutines has been added. The main difference between a subroutine and a formula is that a subroutine is only executed when called (invoked) programmatically from a formula or another subroutine, whereas formulas can not be referenced programmatically (in this respect, a formula is analogous to the "main function", which exists in many programming languages). Up to 20 arguments can be passed to each subroutine. Each argument can be either an expression, or an address of a variable, or a pointer to an array, or a pointer to another subroutine or built-in function (the number and meaning of each formal parameter of a subroutine are defined in the dialog window "Subroutine name and formal parameters"). Recursion is allowed. In order to avoid stack overflow, the subroutine call depth ("nesting level") is limited to 300. Each subroutine can have up to 20 local parameters, which are similar to local parameters of formulas (the local parameters are "visible" only in the corresponding subroutine). Unlike local parameters of formulas, local parameters of subroutines can not be optimized by nonlinear fitting.
5. The "while" loop operator has been added.
6. Support for multi-dimensional arrays has been added. Arrays can have up to 20 dimensions (indices). Arrays in GraphiXT are global objects, which exist independently of user-defined programs. Array element values can be accessed and modified in user-defined programs. Array elements are referenced using the format "a[i1, i2, i3, ...]", where "a" is the array name, and "i1, i2, i3" are the array indices (any expressions can be used instead of them). 8 array data types are supported (64- or 32-bit floating point, and 32-, 16- or 8-bit signed or unsigned integer). However, in all arithmetic operations, the array element values are converted "on the fly" to 64-bit floating-point format.
7. Array properties (name, dimensions and data type) are defined in the "Array" dialog. In addition, that dialog allows initialization of the array elements using an entered expression. That expression can include user-defined identifiers of array indices and references to any programming object defined in the current project, making possible complex initializations.

8. Any user-defined array can be allocated or deallocated at any time. When an array is deallocated, all its elements are lost, but the array object is not deleted. New arrays are not allocated by default. If a user-defined program contains a reference to an array that has not been allocated, then execution of that program would cause a runtime error. This is one of runtime errors handled by GraphiXT (other runtime errors, which are handled by GraphiXT, are listed below).
9. Several built-in arrays have been defined, which allow access to any model data value in user-defined programs (and not just to interpolated model function values corresponding to the current time and current coordinate, as in previous versions of GraphiXT). Similarly, any data point of any free curve can be accessed. All built-in arrays are read-only, i.e., user programs can not modify their elements (those arrays are automatically modified by GraphiXT when the user changes the model data or free curve data).
10. The array viewer has been added. It is based on the same code as function data tables. The array viewer allows direct access and modification of element values of user-defined arrays. In the case of a multidimensional array, any two indices can be chosen to be variable (those two indices are used to number the rows and columns of the array view). The position, range of values of each index, the maximum number of cells that can be shown in an array view, and other options of the array viewer are set in the dialog window "Array view options".
11. The code of the function data tables and the array view has been modified so as to decrease the amount of data associated with each cell of the table. As a result, the total amount of memory required by each function data table or array view has been decreased to less than 1 / 5 of the previous amount.
12. Since an extension of programming capabilities involves an increase of the number of possible programming errors, a lot of effort has been put into minimizing the risk that a runtime error in a user program will crash the GraphiXT executable (causing loss of all unsaved GraphiXT data). Runtime checking for the following types of errors has been added:
 - a) out-of-bounds array indices,
 - b) references to un-allocated arrays,
 - c) abnormally long computation (e.g., due to an infinite loop),
 - d) infinite recursion,
 - e) runtime errors related to limitations of certain built-in functions.

An abnormally long computation of a formula value (such as caused by an infinite loop) is handled by displaying a dialog window with an option to interrupt the computation, when a formula value is computed longer than 2 seconds. The user can either continue waiting (that dialog will be closed automatically after finishing the computation of the current value of the formula), or stop the computation by clicking the button "Stop calculation" at any time. In the latter case, the current value of the computed formula and all subsequent values in the current sequence of values are set equal to the indeterminate number ("-1.#IND").

In the program editor window, the first detected runtime error is shown by automatically scrolling to the corresponding line of code and highlighting it. If the runtime error occurs in a subroutine, then the line containing the corresponding call to that subroutine is highlighted in the code of the calling program, too.
13. For a more convenient access to programming objects, as well as for defining the set of X values of an edited formula, a modeless dialog window "Program editor tools" has been added. It can be hidden or displayed whenever a program editor window is active. In order to display the programming tools dialog, the user has to select the menu command "Programming / Show programming tools dialog", or the corresponding command of the context menu of the program editor.
14. All programming objects (i.e., formulas, subroutines, arrays, parameters and DLL functions) can now be copied and pasted between different projects. Excluding formulas, this is done in the dialog window with the list of objects of that type, using the corresponding buttons or commands of the context menu, or keyboard shortcuts "Ctrl + C", "Ctrl + V", "Ctrl + X" (formulas can be copied and pasted in graph windows, as in previous versions of GraphiXT). A list of subroutines, arrays, parameters or DLL functions can be rearranged by cutting any object and pasting it into a different

position in the same list. In addition, lists of subroutines and arrays have the “Up” and “Down” arrow buttons, which provide a more convenient way to rearrange the list (the keyboard shortcuts are “Ctrl” + “↑” and “Ctrl” + “↓”, respectively). Another difference between lists of subroutines or arrays and lists of parameters or DLL functions is that any change to a list of subroutines or arrays is applied immediately and can not be undone, whereas changes done to a list of parameters or DLL functions can be cancelled.

15. If the loaded plug-in simulates a two-layer system, and layers No. 0 and No. 1 share the same set of coordinate x values, then GraphiXT no longer uses x values belonging to layer No. 0 (i.e., the loaded plug-in is no longer required to allocate the corresponding array). In such a case, the amount of memory that would be taken up by x values of layer No. 0 is no longer taken into account when calculating the total amount of data (which is shown in the dialog window “Time limits and amount of data”), and it is assumed that any reference to an x value belonging to layer No. 0 is equivalent to the reference to the corresponding x value of layer No. 1.
16. In order to decrease the time of drawing a large number of cells, in array views only the visible cells are filled with data, and in data tables only the cells of the visible rows are filled with data (including the cells that belong to non-visible columns). As a result, the data table command “Fit table to contents” has been replaced with the command “Fit table to visible contents”. After selecting that command, all column widths are adjusted to the width of data values in the visible rows. However, after vertical scrolling of the window, some of the data in the newly uncovered cells may be clipped, so that it may be necessary to select the “Fit” command again. *Note:* the “Fit” command is absent in array views. In the array view, all non-fixed columns are forced to be of equal width (a change of one column width causes a corresponding change of the widths of all other columns), and likewise for heights of non-fixed rows.
17. A number of bugs have been fixed.
18. The GraphiXT user’s manual has been updated to reflect the recent changes in the program.

1.20.4.4 (2013-08-10):

1. Nine built-in integration, summation and iteration functions have been added. The built-in function “Int” is used to calculate integrals of user-defined functions over finite intervals. For example, the integral of the function $\exp(-x^2) \cdot \exp(-(t-x)^2)$ with respect to x from 1 to 10 is calculated as follows:

$$\text{result} = \text{Int}(\exp(-x*x) * \exp(-(t-x)^2), x, 1, 10)$$

In the next example, the double integral of a user-defined function FUNC1(a,b) is calculated:

$$\text{result} = \text{Int}(\text{Int}(\text{FUNC1}(a,b), a,a1,a2), b,b1,b2)$$

The built-in function “Inti” is used to calculate integrals of functions over infinite intervals. Example of integration from 5 to $+\infty$:

$$\text{result} = \text{Inti}(\exp(-x*x) * \exp(-(t-x)^2), x, 5, 1)$$

(when integrating from $-\infty$ to a finite bound, the last argument should be -1). Example of integration from $-\infty$ to $+\infty$:

$$\text{result} = \text{Inti}(\exp(-x*x) * \exp(-(t-x)^2), x, 0, 2)$$

(in this case, argument No. 4 must be 2 and argument No. 3 can be any value). The built-in function “Sum” is used for summation. For example, the sum of products of corresponding elements in row 2 of a two-dimensional array A and in column 3 of a two-dimensional array B is calculated as follows:

$$\text{result} = \text{Sum}(A[2,t]*B[t,3], t, 1, \text{Size}(\text{loc}(A[1,1])))$$

Argument No. 1 of all built-in integration, summation and iteration functions defines the expression to be processed, and argument No. 2 defines the integration variable or index of summation, or index of iteration. The expression defined by argument No. 1 stays in scope of the calling program, so that all previously defined variables can be used in it. The integration variable or index of summation takes priority over all other variables (including system variables) and it is only “visible” in the processed expression, so that any identifier can be used in place of that variable. In the previous example, the value of the system variable “t” (the time value) is not affected by the summation.

Detailed descriptions of all built-in integration, summation and iteration functions are given in the updated user’s manual.

2. Calculation of sequences of values (for example, when plotting a formula, or initializing an array, or during nonlinear fitting) has been made faster by reducing the “overhead” associated with calculating each value.
3. All limitations on using the unary minus or unary plus in user programs have been removed. It is now treated according to the generally accepted rules.
4. The maximum allowed subroutine call depth (“nesting level”) has been increased from 300 to 1000.
5. If an expression of a global or local parameter has syntax errors, then after attempting to calculate that parameter, it is assigned the indeterminate value “-1.#IND” (in previous versions, the parameter value remained unchanged).
6. Image of a graph is now copied to clipboard not only in bitmap format, but in Windows metafile format, too.
7. Several bugs have been fixed.
8. The GraphiXT user’s manual has been updated to reflect the recent changes in the program.
2013-08-13:
9. Memory exception handling when allocating arrays or changing array view settings has been added.
10. Runtime errors when attempting to copy a very large amount of data to clipboard have been fixed.
2013-08-14:
11. The meaning of the system variable “NLSF” during nonlinear fitting has been changed: if the fitted dataset belongs to a free curve, then NLSF is equal to the sequence number of that curve in the corresponding graph window, and if the fitted dataset belongs to a model function, then NLSF is opposite to the sequence number of that model function. In addition, six new system variables have been defined for identifying the current data point during nonlinear fitting (see the updated user’s manual for details).
12. A crash after an attempt to open the smoothing dialog when a formula is selected has been fixed.
2013-08-15:
13. Three system variables for use with built-in integration functions have been added: “IER”, “AbsErr” and “nEval”. Those variables are modified by the built-in integration functions and provide additional information about the integration that has been performed (see the updated user’s manual for details).
14. A crash that could happen after a runtime error in a built-in integration function has been fixed.
2013-08-16:
15. A bug that prevented the parameter list from being updated after clicking the button “Calculate All” has been fixed.
2013-08-17:
16. A bug that prevented re-calculation of array-dependent formulas after clicking the button “OK” or “Apply” in the array options dialog window has been fixed.
2013-08-18:
17. The following bug has been fixed: after changing the time slider position or after entering a new time value in the slider bar, the formulas of the current graph that depend on the current time (i.e., on the variable “t”) were not recalculated when the current graph window was of $f(x, t)$ type and the X dataset of the formula was not a temporary set of X values.

2013-08-19:

18. Floating-point exception handling has been added to built-in integration functions.
19. The default maximum number of subdivisions of the integration interval when executing built-in integration functions has been increased from 100 to 200.
20. A bug that prevented update of an array element value during an assignment operation when the right-hand side of the assignment operator was a call to one of “advanced” built-in functions has been fixed.
21. A bug that prevented formulas with syntax errors from being automatically re-compiled and calculated after changing array name or number of indices has been fixed.

2013-08-21:

22. A built-in function “Return(x)” has been added. When it is called, the current formula or subroutine is immediately terminated, returning the value of “x”.

2013-09-19:

23. A bug that sometimes caused incorrect value of model data amount to be shown after changing the number of layers in the simulated system has been fixed.

2013-09-20:

24. When an initialization expression of a parameter or an array is longer than the maximum allowed length (100 characters), a warning message appears.

25. Calculation of the standard error of the mean during elementary statistical analysis (dialog window “Data analysis”) has been corrected. Now, it is calculated by dividing the standard error of single observation by the square root of the total number of observations n (in previous versions, the standard error of single observation was divided by the square root of $n - 1$).

2013-09-21:

26. A bug that sometimes caused incorrect error message after a runtime error has been fixed.

2013-10-06:

27. A bug that could cause a crash during simulation has been fixed. That bug showed up when the simulated system consisted of a single layer and the primary $f(x, t)$ functions were computed with maximum precision (i.e., after checking the checkbox “Primary functions” in the dialog window “Time limits and amount of data”).

28. A bug that sometimes caused incorrect X axis limits after rescale has been fixed. That bug showed up when the graph did not have any model functions, but had formulas whose X datasets coincided with model times values or with model layer node coordinates.

2013-10-27:

29. A bug that could cause a crash when working with an array view containing a large number of rows or columns (several tens of millions) and having a large row height or large column width has been fixed. This fix involves limiting the row height or column width so that the total “virtual” height and width of the table does not exceed $2 \cdot 10^9$ pixels.

2013-11-06:

30. After opening a project file that was saved at a different screen resolution, positions of graphical objects (text labels, legend and free-form lines) in the graph windows are closer to the original positions.

2013-12-02:

31. A bug that could cause a crash after creating a new subroutine or DLL function, when some of the formulas contain references to arrays, has been fixed.

2013-12-07:

32. Runtime errors after changing the number of curves, when there is a data table, which is in “partial” mode and invisible (e.g., minimized or obscured by a maximized graph window), have been fixed.

2013-12-10:

33. Flicker when redrawing graph windows has been greatly reduced.

2013-12-22:

34. Memory leaks have been eliminated.

2014-01-03:

35. The maximum allowed length of identifiers of variables, arrays and functions has been decreased to 50 characters.

36. In the dialog window “Nonlinear least squares fitting”, sizes of three information text boxes and position of the checkbox “vary” have been changed.

2014-01-04:

37. A bug that prevented update of the built-in array XC (abscissas of free curves) after unlinking free curves has been fixed.

2014-01-09:

38. A bug that sometimes caused incorrect redrawing of the legend after a change in the number or meaning of model functions (caused by a change of model parameters in the parameter editor) has been fixed.

2014-01-14:

39. Fixed a crash that was possible after a change in the number of global parameters, when some of the graphs had formulas depending on interpolated model functions (“FT n ”, “FX n ”) or on interpolated free curves (“FC n ”).

40. Fixed a crash that was possible after a change in the number or meaning of model functions, when a graph had formulas depending on model functions, and the data table of that graph was open.

41. Fixed a bug that sometimes prevented update of formulas depending on model functions after a change in the number or meaning of model functions (caused by a change of model parameters in the parameter editor).

2014-01-14:

42. Eliminated a message about compilation error, which showed up while compiling an initialization expression of a parameter or an array when a subroutine editor window was active and the expression required an active graph window.

2014-02-16:

43. Fixed a crash that was possible after attempting to close the nonlinear fitting dialog window when any of the four text input boxes on the right-hand side of the window (current, initial, maximum and minimum values of the current parameter) had input focus.

2014-02-21:

44. Fixed a crash after clicking the “Curve format and name” button in the model function list dialog when that list is empty.

2014-03-01:

45. The comment symbol ‘//’ may now be preceded by code on the same line (as in C++ programs).

2014-03-03:

46. Fixed a crash during nonlinear fitting, when model parameters are being optimized and a function data table is open.

1.21.4.4 (2014-03-10):

1. Added a new menu command “Tools / Options...” and a corresponding toolbar button. After selecting that command, the property sheet window “Project defaults” is opened. That window has eight “tabs”, which are used to select “property pages”, where various default properties can be modified (e.g., default fonts). The changes can be applied either to existing windows of the current project, or to all future windows of the same project (i.e., “project defaults” can be changed), or to all future projects (i.e., “GraphiXT defaults” can be changed). The GraphiXT defaults are stored in the file “GraphiXT.defaults”, which is loaded every time during GraphiXT startup. If that file is absent or if it is empty, then the original defaults will be used.
2. The dialog window with axis options has several new controls, which make it possible to apply the changes not only to the current graph, but also to other graphs or to future projects. In the latter case, the changes will be saved in the mentioned file “GraphiXT.defaults”.
3. When new free curves are created (or pasted, or imported from text files), it is now possible to link them to any existing X dataset. In addition, during data import, it is now possible to select multiple input files.
4. In the fitted dataset options dialog window, a new error-weighting option has been added: the errors can now be divided by another dataset that is linked to the same X dataset as the fitted dataset. Thus, error factors can be specified independently for individual points of the fitted dataset.
5. In the slider bar, added two checkboxes “x-sync” and “x-sync all”, which are used for turning on or turning off the “coordinate synchronization” mode. This is the operation mode when respective X axis limits of two or more coordinate graph windows are equal to each other.
6. Seven new built-in functions have been added: “Select”, “Find”, “Find2”, “Hist”, “Hist2”, “Root”, “Root2”.
7. Three new built-in arrays have been added: “XF”, “FA” and “NF”. They contain formula abscissas, computed formula values and the number of formulas in each graph, respectively. Usage of those arrays in programs is similar to usage of arrays “XC”, “FC” and “NFC”, which refer to free curve data.
8. A new system variable “curForm” has been added. That variable is equal to the sequence number of the currently computed formula in the graph window.
9. The dialog window with options of a formula has a new checkbox that is used to turn on or turn off saving of formula values in the project file. If this option is not selected, then the formula will be re-calculated every time when the project file is opened. The default value of this option can be set using the previously-mentioned menu command “Tools / Options...” (tab “Formulas”).
10. After selecting integration of a formula in the dialog window of elementary data analysis, it is now possible to choose the adaptive integration algorithm instead of the interpolation-based algorithm (the adaptive algorithm is more accurate). If the adaptive algorithm has been selected, then the formula values that are needed for estimation of the integral will be calculated during integration (otherwise, the formula will not be re-calculated, and its plotted values will only be used during integration).
11. Added a text editor menu command that makes it possible to modify some properties of the text editor: font size, saving of editor window size and position, and presence or absence of the status bar (only in code editor). When a text editor window is active, then the font size can also be changed by scrolling the mouse wheel and at the same time keeping the “Ctrl” key down. The default properties of the code editor can be set using the previously-mentioned menu command “Tools / Options...” (tab “Code editor”).

12. In the dialog windows with lists of free curves and formulas, multiple items can be selected, and there are three buttons that perform actions on all selected items at once: “Hide”, “Show” and “Delete”.
13. All dialog windows that contain lists are now resizable. The positions of those lists and other controls in the dialog window depend on its dimensions. The dialog windows with multiple lists are split into vertical or horizontal “panes”, separated by “pane handles”, which can be dragged with the mouse, so that widths or heights of two adjacent lists are changed simultaneously (one is increased, while the other is decreased). The original dimensions of each resizable dialog window and original positions of the mentioned “pane handles” can be restored by selecting the corresponding command of the system menu (the system menu is opened either by right-clicking the title bar of the dialog window, or by left-clicking the window icon, which is in the top left corner of the dialog window).
14. Several bugs have been fixed.
15. The GraphiXT user’s manual has been partially updated to reflect the recent changes in the program.
2014-03-21:
16. Fixed a crash after clicking “OK” in the text label options dialog window.
2014-03-24:
17. The data amount shown in the dialog window “Time limits and amount of data” is calculated including the data of all formulas and “time cross-sections”.
18. Fixed a runtime error that occurred during nonlinear fitting when there were syntax errors in the user-defined formula used to calculate the weight factors.
2014-03-26:
19. Fixed a bug that sometimes caused inconsistent values of X axis limits of new x graphs (i.e., those limits could be different from respective limits of other x graphs even when the checkbox “x-sync all” was checked).
2014-03-28:
20. Fixed an occasional crash after pasting a formula.
2014-03-31:
21. Added an option to insert the input data file name into the names of imported free curves.
22. Fixed a bug that caused incorrect Y values of imported or pasted free curves, when the input data has only one X value and the curves are being linked to an existing X dataset that has more than one element.
2014-04-01:
23. Corrected format of error messages that appear after entering an illegal value into a grid control.
24. Fixed a bug that sometimes caused empty strings to be copied to clipboard when copying x values from a function data table in “partial” mode.
2014-04-02:
25. Added an option to insert the input data file name into the name of the imported X dataset.
2014-04-06:
26. Corrected an occasional error when pasting data into a function data table.
2014-04-11:
27. Added a property page “Number format” to the property sheet “Project defaults”. That property page contains controls for setting the number of significant digits and the use of decimal exponent in the displayed values of array elements, functions, their arguments and parameters. In addition, the parameter value format can be modified in the parameter list by selecting a corresponding command of the context menu.
2014-04-13:
28. Changed positions of several controls in the dialog window “Fitted dataset options”.

2014-04-19:

29. Fixed an occasional crash after changing X values in a function data table, which is in “partial” mode, and then attempting to select a point of a corresponding curve.

2014-04-25:

30. Fixed a bug in the dialog window “Subroutine name and formal parameters” that could cause incorrect interpretation of a list of formal parameters of a subroutine, if that list starts with a comment line.

2014-04-27:

31. Fixed a crash in the dialog window “Subroutines” after attempting to paste and replace a subroutine that is open for editing in a code editor window.

2014-04-28:

32. After a change of a user-defined array, the formulas that are automatically recalculated are not only the ones that depend on that array explicitly, but also all formulas that depend on the built-in array “FA” (computed formula values).

2014-05-02:

33. Fixed a bug that could in rare cases cause the program to stop responding after importing or pasting free curve data, when a function data table is open.

2014-05-05:

34. Inserted the command “Open table window” into the context menu of the formula code editor and into the menu “Programming”, when the formula code editor window is active.

2014-05-10:

35. Fixed a crash that sometimes occurred after a floating-point exception in the user-defined integrand during execution of a built-in integration function.

2014-05-13:

36. Fixed a bug that caused enabling of the checkbox “Apply to the active code window” after checking and then un-checking the checkbox “Apply to all existing programs” in the property page “Code editor” of the property sheet “Project defaults”, even when the current window was not the code editor window.

2014-05-18:

37. Fixed a bug that caused incorrect re-assignment of an X dataset to a formula after creating a free curve and linking it to an X dataset that previously had only one formula linked to it.

2014-05-21:

38. Added a button “Import from a file...” in the dialog window “Array properties”.

2014-05-26:

39. Import of free curve data from multiple text files has been made faster, especially if there are formulas that reference free curve data.

2014-05-29:

40. Fixed a bug that could cause a crash while calculating a built-in integration, summation, iteration or root-finding function, whose first argument (i.e., the processed expression) contained two or more calls to the just-mentioned built-in functions.

41. Fixed a bug that sometimes caused the built-in function Root or Root2 to fail when the processed expression contained calls to built-in integration, summation, iteration or root-finding functions.

2014-06-02:

42. Fixed a bug that sometimes caused a built-in integration, summation or iteration function to return an incorrect value when the first argument of the function (i.e., the processed expression) contained two or more calls to the just-mentioned built-in functions.

2014-06-07:

43. Fixed a bug that sometimes caused incorrect updating of the slider bar after selecting the menu command “Window / Update All Graphs”.

2014-07-29:

44. Improved handling of exceptions in user-defined DLL functions.

2014-07-30:

45. Inserted Section 4.5 “Using DLL functions in programs” into the GraphiXT user’s manual.

2014-08-01:

46. Fixed a bug that prevented the program from detecting an incorrect argument type in a function call when that function was used as a formal parameter of a subroutine.

47. Fixed a bug that sometimes caused a runtime error after deleting a graph window if a formula code editor window was open.

2014-08-05:

48. Fixed a bug that caused the failure to detect an unloadable DLL function during compilation of a user-defined program, when that DLL function was used as an argument of the built-in function “loc”.

2014-08-25:

49. Fixed a bug that caused a crash when calling the built-in function Size2 with the first argument equal to “loc(FC)”, “loc(FA)” or “loc(FX)” and the second argument equal to 3, if the same call with the second argument equal to 1 or 2 would return 0.

2014-08-27:

50. Fixed two bugs that could cause a crash or incorrect reported value of model data amount during nonlinear fitting, when some of varied parameters were model parameters and the time of a single simulation run was extremely short (so that there were several iterations per second).

2015-01-18:

51. Fixed a crash after deleting a row corresponding to a varied parameter from the grid control with the list of parameters in the dialog window “Nonlinear least squares fitting” and then starting the fitting process.

2015-01-23:

52. Fixed incorrect redrawing of graph windows (excluding the current graph window) when margins were changed in the axis properties dialog and option “Apply to all existing graphs” was selected.

2015-01-26:

53. Changed the default error factor (i.e., the square root of the weight factor of squared errors) when calculating the sum of squared errors during nonlinear least squares fitting: it is now equal to 1 (in previous versions, it was equal to 100 divided by the average absolute value of Y in the fitting range).

2015-01-27:

54. Fixed the failure to restore initial numbering of curves in a graph window after creating a formula and then closing the code editor window without saving the changes. [Among other things, this bug would sometimes cause the message “Encountered an improper argument”, which could subsequently appear after clicking an item of the legend.]

2015-02-23:

55. Fixed a bug that sometimes prevented an automatic update of the slider bar after changing the time multiplier in the X axis options dialog.

2015-02-24:

56. Fixed a crash after clicking the button “Insert before” in the subroutine list dialog window.

2015-02-27:

57. Fixed an occasional crash after an attempt to close a subroutine code editor window.

58. Fixed a bug that prevented automatic recalculation of formulas after changing X dataset properties in the dialog window “X datasets”.

2015-04-07:

59. Fixed a bug that caused incorrect recalculation of the x or y coordinate of the reference point of a text label or a free-form line after entering its value in the text box “ $X =$ ” or “ $Y =$ ” of the options dialog, if the X or Y axis multiplier was not equal to 1.

2015-04-09:

60. Fixed a bug that sometimes caused incorrect rescaling of the time axis after importing model data from text files.

2015-07-28:

61. Fixed a bug that sometimes caused a crash after pasting data into a function data table, which is in “partial” mode.

2015-09-14:

62. Fixed a bug that caused a failure to redraw automatically array view windows after initializing an array, whose initialization expression contains a call to a subroutine, which modifies array data.