



EXTEND™

ReadMe

This ReadMe contains useful information about:

Extend Features

Version 5 Upgrade Information

The Suite Products and the SDI Industry Module

Extend User Forum

Imagine That! Web Site

Purchasing an Extend Product

Training and Technical Support

Release Notes

Extend Features

To see a comprehensive list of Extend features, please go to:

www.imaginethatinc.com/extendfeatures.html

Version 5 Upgrade Information

If this is an upgrade from an earlier version of Extend, please go to

www.imaginethatinc.com/v5.html to see a list of what's new in v5.

The Suite Products and the SDI Industry Module

The two Suite products are perfect for those who want total simulation project support. Extend Suite and Industry Suite feature Proof Animation (for realistic, professional-level animations) and Stat::Fit (to statistically fit your data to the most useful distribution). Extend Suite bundles Extend+BPR+Manufacturing with Proof Animation and Stat::Fit; Industry Suite adds Industry to the Extend Suite bundle .

The SDI Industry module (bundled in Extend+Industry and Industry Suite) provides a complete data management system for your simulation models. By separating your data from your model, the system enables fast scenario implementation, flexible analysis and improved project management. As a bonus for high speed/ high volume processes, SDI Industry also provides a complete set of flow blocks, which give you the option of modeling material in terms of rates instead of items.

For more information, go to www.imaginethatinc.com/products.html

Imagine That, Inc.

6830 Via Del Oro, Suite 230 • San Jose, CA 95119 USA

408-365-0305 • 408-629-1251 (fax) • extend@imaginethatinc.com

www.imaginethatinc.com

Extend User Forum

Talk directly to other simulationists in the Extend E-Xchange, the virtual meeting room where you can exchange ideas, discuss simulation techniques, and post issues.

www.extendusers.com

Imagine That! Web Site

Don't forget to check the Extend web site www.imaginethatinc.com for up-to-date news and information on Extend...new releases, additional modules, new or enhanced blocks, sample models from Imagine That! and Extend users, frequently asked questions, conferences and shows we'll be attending, and more.

Purchasing an Extend Product

For information on purchasing additional Extend products or to add an Extend module to your existing Extend product please go to www.imaginethatinc.com/order.html or contact Imagine That! at sales@imaginethatinc.com or 408-365-0305.

Extend Training and Technical Support

Go to www.imaginethatinc.com/support.html to obtain product updates, download additional example models, or see lists of training locations and frequently asked questions. Or contact us with your support or training questions at (telephone) 408-365-0305, (fax) 408-629-1251, or support@imaginethatinc.com

Release Notes

Notes for Block Developers

The Extend version 5 compiler has many optimizations and bug fixes. For optimal performance, we recommend that you recompile any blocks built in earlier versions. Before you do this, however, please note that version 5 is not backward compatible and blocks compiled in v5 cannot be read by earlier versions of Extend. Also, follow normal software-handling precautions and make a backup copy of your libraries before you recompile them in Extend 5.

General Enhancements

AVI example files (Windows version only)

A new Movies folder in the Extend5 folder contains AVI files that illustrate specific Extend products, such as Industry (bundled in the Extend+Industry and Industry Suite packages), and Proof Animation and Stat::Fit (bundled in the Extend Suite and Industry Suite packages). Launch these AVI files by double-clicking directly, or open them using the buttons in the Demo.mox example file.

Graphics import format (Windows version only)

Prior to Extend 5.0.3, graphics imported through the clipboard or pictures from the extensions folder could be distorted. Extend now searches for and will give preference to using a BMP file. It will only use a WMF file if a BMP file is not available. For the best graphics results, it is recommended that you use BMP files rather than WMF files.

Changes From the Printed Documentation

Antithetic Random Variates • User's Guide, page E207

This is a new checkbox in the Random Numbers tab of the Simulation Setup dialog. When checked, Antithetic Random Variates causes all distributions to be generated based on the complement of the uniform distribution (e.g. $u = 1-u$). This is used in variance reduction to find the mean value of the output of a model using fewer runs. Using a seed, the model is run with non-antithetic variates and then is run using the same seed with antithetic variates. When the two run outputs are averaged, the result might be closer

to the expected mean than from two runs averaged using non-antithetic variates. How effective this technique is depends on the type of model and is not easily ascertained. The Evolutionary Optimizer block has options to use antithetic variates.

🔥 [ExecuteMenuCommand](#) • Programmer's Reference, page P158

The Description should read: See Appendix C for a list of menu command numbers.

New and Revised Menu Commands

🔥 [File menu, Launch StatFit command](#) • User's Guide, page A3

This new command opens the Stat::Fit[®] application for Windows. This command is only enabled if Stat::Fit has been installed within the Extend folder. Stat::Fit is a distribution-fitting application included in the Extend Suite and Industry Suite packages; it can also be purchased separately from Geer Mountain Software Corp.

🔥 [Edit menu, Preferences command](#) • User's Guide, page A8

The Model tab in the Preferences dialog now includes a field where you can enter the path name of a model file. The specified model will automatically be opened when Extend is launched.

New and Revised Functions

🔥 [OLEDispatchGetDoc\(integer IDispatchHandle, string returnDoc\[\], integer dispID, integer which\)](#) • Integer.

Programmer's Reference, page P107

This function is a Dispatch handle version of the OLEGetDoc function on page 108. For a description of the IDispatchHandle argument, see the OLEDispatchGetHelpContext function on page P107.

🔥 [ODBCColumns2\(integer hdbc, string catalog, string schema, string table, string column\)](#) • Integer.

Programmer's Reference, page P113

This function works the same as the ODBCColumns function defined on page P113 except that it adds four string arguments. See your ODBC documentation (SQLColumns) for additional information about the meaning and use of these arguments. Please note that the arguments are strings, so you cannot pass in the NULL values that are defined in the SQLColumns specification. We've defined an empty string "" to be interpreted as a NULL, so you should just use empty strings for unused arguments.

🔥 [ODBCExecuteArray\(integer hdbc, string array\[\]\)](#) • Integer.

Programmer's Reference, page P114

See also the description for ODBCExecuteQuery on page 114. This function allows a query that contains more than 255 characters, by allowing you to pass in an array of strings instead of just one string.

🔥 [ODBCKeyword\(string word\)](#) • Integer.

Programmer's Reference, page P115

Returns a TRUE value if the string word is an ODBC reserved keyword. Otherwise returns a FALSE value.

🔥 [ODBCSetRowsType\(integer hdbc, string tableName, string colName, string IDName, string y\[\], string x\[\], integer varTypeX, integer varTypeY\)](#) • Integer.

Programmer's Reference, page P115

See also the description for the ODBCSetRows function on page 115. Whereas the SetRows function defaults the types of the values of both string arrays to SQL_CHAR, the SetRowsType function allows you to specify the types of the variables in the y and x arrays. Type values are:

SQL_UNKNOWN_TYPE	0
SQL_CHAR	1
SQL_NUMERIC	2
SQL_DECIMAL	3
SQL_INTEGER	4
SQL_SMALLINT	5
SQL_FLOAT	6
SQL_REAL	7

SQL_DOUBLE	8
SQL_DATETIME	9
SQL_VARCHAR	12
SQL_TYPE_DATE	91
SQL_TYPE_TIME	92
SQL_TYPE_TIMESTAMP	93

🔥 [FindInHierarchy\(string FindBlockName, string FindBlockLabel, string FindDialogName, long FindDialog\)](#) • Integer. Programmer's Reference, page P129

The correct description for this function is:

This function is used by several blocks, including the Catch and Throw blocks in the Discrete Event library, to locate the corresponding block associated with the current block. (For example, to find a Catch corresponding to a Throw, and vice versa.) Please see the Catch and Throw blocks for an example of how to use this function. The function returns the block number of the resulting block found; it returns a -1 if no matching block is found.

FindBlockName is the name of the block to be found (e.g. 'Catch', or 'Throw')

FindBockLabel is the block label of the block to be found.

FindDialogName is the name of the dialog item you wish to search for. The FindDialogName field should be filled with the dialog item name, a colon, and then the value of the dialog item that you wish to search for. For example *Item:54* will search for the presence of a dialog item with the name "item", and the value "54".

FindDialog takes the following values:

- 0: just check the block name, and the block label.
- 1: just check the block name and the dialog item name and value.
- 2: check the name, label, and dialog item

🔥 [GAInitializing\(integer ArrayIndex, integer Initializing\)](#) • Integer. Programmer's Reference, page P151

A new Global Array function, it sets the initializing flag for the specified global array. The flag determines if the array is automatically initialized during initsim of a model run or not. The Initializing flag takes the following values:

- 0: don't initialize (default)
- 1: initialize to 0
- 2: initialize to blank (real numbers only.)

🔥 [GAGetInfo\(integer ArrayIndex, integer Which\)](#) • Integer. Programmer's Reference, page P151

This new Global Array function just returns the value of some of the Global Array flags for the specified array. Values for "Which" are:

- 0 - non saving
- 1 - initializing

🔥 [IsLibEnabled\(integer Library\)](#) • Integer. Programmer's Reference, page P160

The Library value is 4 for Industry.

🔥 [GetLibraryVersion\(integer BlockNum\)](#) • String. Programmer's Reference, page P168

This new function returns the short version string for the library that includes the block specified by BlockNum.