

RELEASE NOTES

MATRIXx™

Version 7.1

These release notes contain system requirements, and information about new features included in MATRIXx 7.1, the Xmath, AutoCode, and SystemBuild Modules.

Platforms, Requirements, and Deliverables

MATRIXx runs on Windows 2000/NT/XP and Solaris 7 and 8.

The MATRIXx 7.1 product CD includes Altia Design 5.28, Altia FacePlate 5.28, and the MATRIXx 7.1 Bookshelf in portable document format (PDF) .

The *MATRIXx Help* requires an Internet Browser that can run JavaScript. Microsoft Internet Explorer or Netscape Navigator can be used.

Documentation

The MATRIXx Bookshelf is part of the MATRIXx 7.1 CD and can be invoked from the Xmath **Help** menu or by opening the `$MATRIXX/xmath/help/bookshelf/matrixx.pdf` file in Adobe Acrobat Reader. The MATRIXx Bookshelf displays a structured list of MATRIXx manuals available in PDF. To view PDF documents, you must have Adobe Acrobat Reader Version 5.0.5 or later with Search and Accessibility to use the search functionality. You can download the reader from www.adobe.com.

The MATRIXx Bookshelf PDF includes documentation for all MATRIXx products including Xmath, SystemBuild, AutoCode, and DocumentIt. It also includes FLEXIm End User Manual from Macrovision Software and animation documentation from Altia.

Hardcopy Documentation

PDF files can be printed on most printers. For more information on Adobe PDF format or related products, refer to the Adobe Web site at www.adobe.com. *MATRIXx Help* can be printed from the Internet Browser.

MATRIXx Help

MATRIXx provides a hypertext markup language (HTML) Help system. The *MATRIXx Help* is a self-contained system with multiple hypertext links from one topic to another. To start the *MATRIXx Help* and get instructions on viewing, navigating, and printing topics, type `help matrixx` in the Xmath Commands window or direct your Internet Browser to `file://$MATRIXX/xmath/help/help.html`.



Note You will have to enter the full path to where MATRIXx is installed instead of using the `$MATRIXX` environment variable as shown above. Alternatively, you may use the Internet Browser's Browse feature when selecting **File>Open** in the browser.

What's New in MATRIXx 7.1

The following features have been added since the release of MATRIXx 6.32:

- Xmath graphics function `plot2d()` providing access to `plot()` and `uiplot()` features plus multiple windows and data viewing.
- Xmath PC Debugger window allows you to interactively debug, edit, save, and re-run MathScript functions and commands.
- Xmath PC command line now echoes the last invalid Xmath command entered, highlighting the invalid part of the command. This makes it easier to correct a mistake and re-enter the command.
- Configuration management interface to commercial file and version control systems.
- Enhanced SystemBuild Catalog Browser provides a Catalog view and File view. The File view folders represent all currently opened MATRIXx data files and when selected, lists the catalog objects and Xmath partitions associated with each file. The Catalog view adds new folders for Model, Variables, User Types, and Xmath Partitions.
- IfThenElse blocks now have output ports (pins) and a prolog section.
- Block diagram enhancements, including allowing pins to enter or exit from any side, flexible block name positioning, and expanded character set for labels.

- State Transition Diagram (STD) enhancements allow Input and Output labels to be defined and displayed on STD diagrams. The STD properties dialog box now has Input, Output, and Comment tabs.
- Optimized and restructured fixed-point functions achieving more than 50% size reduction in some applications.
- Code generation template for the Wind River RTOS: VxWorks 5.4.
- Improvements to AutoCode and DocumentIt.
- FLEXlm License Management Software was upgraded from 6.0k to 8.4a. FLEXlm 8.4a has a license borrowing feature enabling MATRIXx to be used on a portable computer off-site with a borrowed license.
- MATRIXx 7.1 replaces the FLEXlm `mtx1md` license daemon in the `$MTXHOME/ISILM` directory with the National Instruments FLEXlm `nilm` license daemon residing in the new `$MTXHOME/NILM` directory.

New or Improved Features

Enhanced Catalog Browser

The Catalog Browser has an enhanced Catalog pane on the left side of the SystemBuild Catalog Browser, which has the following new sections under Main—Model, Variables, and UserTypes. A new Xmath Partitions section was added. It can be expanded to display a folder to each Xmath Partition. Selecting an Xmath Partition causes the Contents pane (right side) to display the name, size, and comment for all of the variables in the partition.

The model hierarchy tree which previously displayed under the SuperBlock folder is now displayed under the new Model folder and will display nested State Diagrams.

The folders for SuperBlocks, State Diagrams, DataStores, Components, Variables, User Types, and Xmath Partitions each display a non-hierarchical list of elements.

The Variables folder acts similarly to the DataStores folder. When selected, the Contents pane displays all of the Global Variables defined at this scope of the catalog.

The UserType folder, when selected, displays in the Contents pane all the User Types defined in the Main SystemBuild Catalog. In previous releases, the User Types were stored in the Xmath partition `_usertypes` and created using the `usertype` Xmath GUI. Since the release of MATRIXx 7.0, User Types are now stored in the SystemBuild Main Catalog and can be created using the old `usertype` Xmath GUI, `SBA createusertype` command or by selecting **File»New»UserType** in the Catalog Browser.

State Transition Diagram Enhancements

Transition conditions displayed in the STD diagram can now optionally select $\cup n$ (for example, \cup , $\cup 2$, $\cup 3$), InputLabels, or InputSignals in the transition expression. The format of the outputs displayed on Bubbles (Moore) and transitions (Mealy) in the STD diagram can now optionally select ChannelNumbers, $\forall n$ (for example, \forall , $\forall 2$, $\forall 3$), or OutputLabels.

The InputLabels and OutputLabels are entered in the new Input, Output, and Comment tabs of the STD Properties dialog box. The InputLabels and OutputLabels entered in the STD Properties dialog box can be propagated to STD references using the PropagateLabels checkbox in the STD reference block dialog box Display tab, similar to SuperBlock references.

The STD Properties dialog box Input tab has the following fields: InstancePath, InputDisplay, Higher Level InputSignal(s), and Input Label(s). The InstancePath fields display a list-box containing SBname/blkId for all locations this STD is referenced. Changing the InstancePath updates the Higher Level Input Signal(s) column to display the signal names of the inputs connected to the STD reference.

The STD Properties dialog box Output tab has the following fields: OutputDisplay, OutputLabel(s), Activating Bubble(s), Activating Transition(s), Deactivating Bubble(s), and Deactivating Transition(s). The activating and deactivating columns show which BubbleIds or Transitions {FromBubbleId, priority} set or reset each STD output.

Improved Block Face and Interfaces

SystemBuild has interfaces that allow pins to enter or exit from any side (top, left, right, or bottom) for input or output pins. You can define a block input face and a block output face independently.

You can rotate block names by using the `rotation` keyword. Block and block name rotation are independent. You can use the new `direction` keyword to toggle the definitions of the block input and output faces.

Block Diagram Enhancements

SystemBuild 7.1 includes 4-way input/output face entry/exit. You can select a left (existing), right, top, or bottom entry or exit for a block from the Inputs or the Display tab.

Configuration Management

The configuration management (CM) feature of SystemBuild enhances the ability to manage data files in Rational ClearCase 3.2.1 or later, Merant PVCS 6.6 or later, and **(Windows)** Microsoft Visual SourceSafe 6.0 or later. CM, accessed from the Versioning menu and the FileView tab of the Catalog Browser, has the following features:

- Saves catalog items into their home files.
- Provides a file view for viewing currently loaded files.
- Tracks status on new, modified, moved, overwritten, and deleted items.
- Provides a seamless interface to basic CM operations from within SystemBuild.

For detailed information about the above and other CM features, refer to the *SystemBuild User Guide*.

Extended Character Set for Labels

SystemBuild supports an extended character set for block output labels and SuperBlock external input labels. This extended character set includes the !"#%&'*+,-./<=>?@^ characters and the rest of the standard ASCII 128 character set, except it excludes all control characters (ASCII 1–32) and () {} [] ; \ ` | ~ &.

IfThenElse Block

The IfThenElse block has been improved with output ports (pins) and a Prolog section.

The IfThenElse block output ports are available from the face of the first condition block in the IfThenElse block chain. You can connect to these ports as if they were the output ports of any other standard block. Output from each of the block sections is connectable to the output ports the same way the output of the content of a SuperBlock is connected to its external output. With this capability, you no longer need to use variable blocks and the Sequencer with the IfThenElse block.

You can define default values for each output channel in the Prolog section. SystemBuild executes the Prolog section before any other IfThenElse sections. This guarantees that all of the output has been defined.

If any of the output of the IfThenElse block is not connected in each section of the block structure, some output may not be assigned if that output is not defined in the branch of the block that is being executed. This can cause some of the output to be undefined unless you use the Prolog section.

This section appears as the uppermost section in the block and is defined by typing `prolog` in the Code tab of the IfThenElse Block Properties dialog box.

Signal Splitting

Signal splitting is now marked in a diagram. Signal splitting occurs when one output is connected to the input of multiple blocks. These markers are placed automatically on the SystemBuild diagram whenever two or more connection lines that share the same path from the same source branch off.

AutoCode and DocumentIt

AutoCode 7.1 provides the following new features:

- Fixed-point libraries (standalone utilities and more source code files)
- Variable step-size solver
- Code Generation template support for VxWorks 5.4
- Name mangling report
- `makefile` generation and the `acmake` command
- The AutoCode SDK interface is now part of the AutoCode product

Xmath

Xmath 7.1 introduces `plot2d()`, a graphics function that provides a shortcut to PGUI `uiPlot()` features for users familiar with the `plot()` function. `plot2d()` accepts the parameters and keywords of both `plot()` and `uiPlot()`. By converting `plot()` function calls to `plot2d()`, you can employ all `uiPlot()` functionality, including:

- **Multiple Plot Windows**—You can create plots on multiple windows from the Xmath Commands window and address plot commands to any given existing plot window.
- **Enhanced Command Capability**—You can specify and update all plot attributes with keywords issued from the Xmath Commands window.
- **Interactive Data Viewing**—With the mouse, you can interactively display the *x* and *y* values of points along any plot line.
- **Multiple Y Axes**—You can display multiple Y axis scales on a plot.
- **Improved Default Background and Line Colors**—New default background and line colors improve the visual presentation of plots.
- **Improved Plot Attributes**—Grid line and tick mark spacings, as well as text font sizes, are determined relative to the size of the plot.
- **Plot Legend Placement**—You control placement of plot legends with `plot2d` command options.

For more information about Xmath graphics, refer to the *Xmath User Guide* and the *uiPlot* and *plot2d* topics of the *MATRIXx Help*.

Xmath Debugger Window

The Xmath Debugger Window is now available for Microsoft Windows and allows you to interactively debug MathScript functions and commands. All debugging functions can be controlled from it. The PC Debugger Window has the same layout as the UNIX version. The editable source code is displayed on the top with the error message area in the middle and an array of buttons for controlling the debugging session on the bottom. The line that is about to be executed is highlighted, unless there are errors in the function, in which case the highlighted line points to the error.

Breakpoints can be set by moving the cursor in the source code to the desired line of code and clicking the **Set Break** button.

Watchpoints can be set by using the mouse to highlight the name of the variable you want to watch and clicking the **Set Watch** button.

One use case for the Debugger Window is when an error occurs in a function. In that case, the Debugger Window displays highlighting the line in error. The function source code can be modified and saved directly in the Debugger Window. By clicking the **Rerun** button, the function can be rerun using the same input arguments.

Additional MATRIXx Information

For more information on MATRIXx or other National Instruments products, refer to the National Instruments Web site at ni.com.

Refer to the MATRIXx product documentation or visit ni.com/support for more information.

AutoCode™, DocumentIt™, MATRIXx™, National Instruments™, NI™, ni.com™, SystemBuild™, and Xmath™ are trademarks of National Instruments Corporation. Product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

© 2003–2004 National Instruments Corp. All rights reserved.



371151A-01

Apr04