

LabVIEW™ 升级说明

8.0 版

本升级说明描述了如何将运行于 Windows、Mac OS 和 Linux 平台上的 LabVIEW 升级至 8.0 版的详细步骤、升级时可能碰到的各种问题以及 8.0 版的最新功能。

关于 LabVIEW 8.0 功能、LabVIEW 编程理论、编程分步指导以及 VI、函数、选板、菜单和工具等参考信息见 *LabVIEW 帮助 (LabVIEW Help)*。*LabVIEW Help* 中也列出了由 National Instruments 提供的各种 LabVIEW 文档资源。请在菜单中选择 **Help»Search the LabVIEW Help** 打开 *LabVIEW Help*。

目录

升级至 LabVIEW 8.0.....	2
转换 VI.....	2
升级工具包、仪器驱动程序和附加工具.....	3
升级其它 NI 软件.....	3
升级 LabVIEW 前期版本.....	4
升级和兼容性问题.....	5
升级 LabVIEW 7.x.....	5
升级 LabVIEW 6.x.....	20
升级 LabVIEW 5.x.....	24
升级 LabVIEW 4.x.....	25
升级 LabVIEW 3.x 和前期版本.....	27
LabVIEW 8.0 功能和更改.....	27
激活 LabVIEW 许可证.....	27
启动 LabVIEW.....	27
登录或注销 LabVIEW.....	28
域帐号管理器.....	28
新建对话框的改进.....	28
RT、FPGA 和 PDA 终端.....	28
LabVIEW 项目.....	29
LabVIEW 项目库.....	34
利用共享变量实现实时数据共享.....	35
创建源代码发布.....	37
Application Builder 前期版本更改.....	37
新选板布局.....	39
菜单重组.....	43
使用源代码控制.....	43

VI 和函数的改进	45
VI 层次结构窗口改进	57
前面板改进	57
LabVIEW MathScript（仅适用于 Windows）	66
矩阵数据类型	67
选项对话框改进	69
另存为和保存为前期版本对话框	69
文件大小改进	69
在各个版本的 LabVIEW 中使用动态链接库	70
设置窗口运行位置	70
设置运行语言选项	70
VI Server 改进	71
类浏览器窗口	72
使用 .NET 集合 (.NET Assembly)	72
仪器驱动查找工具	73
仪器驱动 VI 向导	73
仪器驱动项目向导	73
使用和调试可重入 VI	74
在 VI 运行后创建探针	74
记录多个终端中的 VI 信息	75
64 位整数数据类型	75
NI Spy	76
查找和替换功能	76
自定义键盘快捷键	76
打印对话框改进	76
编辑 VI 图标	76
文档改进和修订	77
NI Example Finder 改进	80
LabVIEW 8.0 其它功能和更改	81

升级至 LabVIEW 8.0

如果从 LabVIEW 前期版本进行升级，请先阅读本文档的转换 VI、升级工具包、仪器驱动程序和附加工具及升级和兼容性问题等章节。

转换 VI

在 LabVIEW 8.0 中，打开由 LabVIEW 4.0 或更高版本所保存的 VI 时，LabVIEW 8.0 会自动转换并编译该 VI。必须在 LabVIEW 8.0 中保存该 VI，否则每次打开该 VI 都需进行转换，并需占用额外内存资源。

对于任何含有未保存改动的 VI，它的运行性能可能会大大降低，因为需要在运行时重新编译该 VI。关于性能和内存问题的详细信息见 *LabVIEW Help*。



注

LabVIEW 前期版本无法加载由 LabVIEW 8.0 保存的 VI。如需在 LabVIEW 7.1 中运行该 VI，可选择 **File»Save for Previous Version** 进行保存。如需由 LabVIEW 7.1 及前期版本运行该 VI，请在使用 LabVIEW 8.0 保存该 VI 之前进行备份。

如计算机无法提供足够的内存一次转换所有 VI，也可选择分层次逐步转换 VI。请检查需转换的 VI 的层次结构，先打开并保存处于较低层次的子 VI，然后逐步转换较高层次的 VI，最后再打开并转换顶层 VI。也可选择 **Tools»Advanced»Mass Compile** 转换整个 VI 目录，但批量编译 (Mass Compile) 将以字母顺序转换 VI 目录或 LLB 中的 VI。如转换过程先碰到较高层次 VI，批量编译所需的内存可能与先打开该 VI 所占用的内存相等。

如需监控 LabVIEW 的内存使用情况，请选择 **Help»About LabVIEW** 显示正在使用的内存总量。

升级工具包、仪器驱动程序和附加工具

请确保您所使用的工具包或附加工具版本与 LabVIEW 8.0 兼容，需要在安装 LabVIEW 8.0 后将这些附加软件重新安装至 LabVIEW 8.0 目录中。如果在 LabVIEW 前期版本中已安装过该工具包，那么您可能首先需要将其卸载。关于软件安装的详细信息见 LabVIEW 工具包或附加工具的相关文档。

同时在 LabVIEW 8.0 中使用现有工具包、仪器驱动程序和附加工具 VI 之前，必须将其进行批量编译。关于 LabVIEW 8.0 中批量编译 VI 的详细信息见本文档的转换 VI 一节。

对于以下工具包、仪器驱动程序和附加工具，必须升级或下载 LabVIEW 8.0 所要求的版本。

- 如已安装 LabVIEW Application Builder，请升级至 LabVIEW Application Builder 8.0。LabVIEW 8.0 专业版开发系统已包括 Application Builder 8.0。关于安装 LabVIEW Application Builder 的详细信息见 `labview\readme` 目录中的 *LabVIEW Application Builder 自述文件*。
- 在 LabVIEW 8.0 中必须使用 VI Analyzer 1.1。请登录 National Instruments 网站 ni.com/info 并输入信息代码 `exd8yy`，访问升级配置指南并 VI Analyzer 1.1。
- 在 LabVIEW 8.0 中使用 Internet Toolkit 6.0 必须先下载一些附加 VI。关于附加 VI 的下载信息，请登录 National Instruments 网站 ni.com/info 并输入信息代码 `exkrkb` 进行查询。
- HP/Agilent 34401A 数字万用表 (DMM) 的驱动程序现更类似于 National Instruments DMM 模板驱动程序，该驱动程序与 LabVIEW 7.x 和前期版本所使用的 HP34401A 驱动程序并不兼容。如需兼容 LabVIEW 7.x HP34401A 驱动程序，请登录 National Instruments 仪器驱动程序网站 ni.com/idnet 下载相应驱动程序。

升级其它 NI 软件

在 LabVIEW 8.0 中必须使用 NI TestStand 3.5 或更高版本。请登录 National Instruments 网站 ni.com/info 并输入信息代码 `exd8yy`，访问升级配置指南并下载 NI TestStand 3.5 或更高版本。

在 LabVIEW 8.0 中必须使用 Logos 4.6 或更高版本。关于 Logos 最新版本的下载信息，请登录 National Instruments 网站 ni.com/info 并输入信息代码 `exymjt` 进行查询。

升级 LabVIEW 前期版本

LabVIEW 新版本将安装在不同目录中，因此在同一计算机中将 LabVIEW 升级至最新版本并不会影响前期版本的使用。LabVIEW 5.x 以及前期版本均安装在 `labview` 目录中。LabVIEW 6.0 以及更高版本安装在 `labview x.x` 目录中，其中 `x.x` 表示版本号。无需卸载 LabVIEW 前期版本就可以安装 LabVIEW 8.0。

如需替换现有的 LabVIEW 版本，请先卸载现有版本，然后运行 LabVIEW 8.0 安装程序，并将默认安装路径设置为被替换的 LabVIEW 版本的 `labview` 安装路径。

(Windows) 也可通过“控制面板”的“添加 / 删除程序”卸载现有版本 LabVIEW，并升级至 LabVIEW 8.0。卸载程序不会删除 `labview` 目录中的文件。



注

卸载或重新安装 LabVIEW 时，安装程序会卸载 `vi.lib` 目录中所有 `.llb` 文件（包括保存在 `.llb` 文件中的所有 VI 和控件）。将 VI 和控件保存至 `user.lib` 目录并手动添加到**控件 (Controls)** 和**函数 (Functions)** 选板中。

使用 LabVIEW 前期版本的环境设置，只需复制该版本对应 `labview` 目录中的 LabVIEW 参数配置文件。



注意

如果将 LabVIEW 8.0 的参数配置文件替换为前期版本的参数配置文件，可能会覆盖前期版本发布后新增加的 LabVIEW 的特定设置。

安装 LabVIEW 8.0 后，请将 LabVIEW 参数配置文件复制到 LabVIEW 8.0 目录中。

(Windows) LabVIEW 将参数保存在 `labview.ini` 文件中。

(Mac OS) LabVIEW 将参数保存在主目录下 `Library:Preferences` 文件夹的 LabVIEW Preferences 文件中。

(Linux) LabVIEW 将参数保存在主目录的 `.labviewrc` 文件中。

要使用 LabVIEW 前期版本 `user.lib` 目录中的文件，只需复制前期版本 `labview` 目录中的文件。并在安装 LabVIEW 8.0 后，将文件复制到 LabVIEW 8.0 的 `user.lib` 目录中。

升级和兼容性问题

关于不同版本 LabVIEW 的升级和兼容性问题的详细信息见以下章节。

升级 LabVIEW 7.x

从 LabVIEW 7.x 升级至 LabVIEW 8.0 时，可能会遇到以下兼容性问题。

支持平台

LabVIEW 8.0 所支持的平台有如下改动：

- LabVIEW 7.1 及更高版本不再支持 Windows Me/98/95。
LabVIEW 8.0 不再支持 Windows NT。
- LabVIEW 8.0 可支持 Windows XP x64 操作系统，但目前只有少数 NI 驱动程序适用于 XP x64。
- LabVIEW 8.0 不再支持 Mac OS X 10.2 或前期版本。
- LabVIEW 8.0 不再支持 Sun Solaris。

系统要求

运行 LabVIEW 7.x 至少需要 128 MB 内存，但 NI 建议您使用 256 MB 内存；运行 LabVIEW 8.0 至少需要 256 MB 内存，但 NI 建议您使用 512 MB 内存。

LabVIEW 7.x 所需屏幕分辨率为 800×600 像素，但 NI 建议将屏幕分辨率设为 1024×768 像素；LabVIEW 8.0 需要将屏幕分辨率设为 1024×768 像素。

(Windows) LabVIEW 7.x 要求使用 Pentium III、Celeron 600 MHz 或更高频率处理器（或等效处理器），但 NI 建议您使用 Pentium 4 或等效处理器。LabVIEW 8.0 要求使用 Pentium III、Celeron 866 MHz 或更高频率处理器（或等效处理器），但 NI 建议您使用 Pentium 4/M 或等效处理器。

(Linux) LabVIEW 7.x 要求使用 Pentium III、Celeron 600 MHz 或更高频率处理器（或等效处理器），但 NI 建议您使用 Pentium 4 或等效处理器。LabVIEW 8.0 要求使用 Pentium III 或 Celeron 866 MHz、更高频率处理器（或等效处理器），但 NI 建议您使用 Pentium 4/M 或等效处理器。

(Windows) LabVIEW 7.x 基本安装和完整安装分别至少需要 130 MB 和 550 MB 磁盘空间。LabVIEW 8.0 基本安装和完整安装分别至少需要 900 MB 和 1.2 GB 磁盘空间。

(Mac OS) LabVIEW 7.x 基本安装和完整安装分别至少需要 280 MB 和 350 MB 磁盘空间；LabVIEW 8.0 基本安装和完整安装分别至少需要 500 MB 和 600 MB 磁盘空间。

(Linux) LabVIEW 7.x 基本安装和完整安装分别至少需要 200 MB 和 300 MB 磁盘空间。LabVIEW 8.0 基本安装和完整安装分别至少需要 400 MB 和 550 MB 磁盘空间。

(Linux) LabVIEW 7.x 需安装 GNU C Library (glibc) 2.1.3 或更高版本，但 NI 建议安装 GNU C Library 2.2.4 或更高版本。LabVIEW 8.0 需安装 GNU C Library 2.2.4 或更高版本。

(Linux) LabVIEW 7.x 可在以下系统中运行：Red Hat Linux 7.0 或更高版本、Mandrake Linux 8.0 或更高版本、SuSE Linux 7.1 或更高版本以及 Debian Linux 3.0 或更高版本。LabVIEW 8.0 可在以下系统中运行：Red Hat Enterprise Linux WS 3 或更高版本、MandrakeLinux/Mandriva 10.0 或更高版本、SuSE Linux 9.1 或更高版本。

自定义选板视图

LabVIEW 8.0 不再支持自定义选板视图 (custom palette views)。现在无需使用自定义选板视图即可编辑选板。关于 LabVIEW 8.0 中选板改动的详细信息见本文档的选板编辑改进一节。

VI 和函数行为更改

LabVIEW 8.0 中包括以下 VI 和函数的行为更改。

.NET VI 和应用程序

必须安装 .NET Framework 1.1 Service Pack 1 或更高版本才可以在 LabVIEW 8.0 中使用 .NET 函数和应用程序。安装 .NET Framework 1.1 Service Pack 1 之前必须删除 Microsoft .NET Framework 1.1 Hotfix KB886904。

如需加载 LabVIEW 7.x 所保存的 .NET VI，即使该 VI 所引用的资源文件与该 VI 在同一目录中或者已经在 LabVIEW 7.x 中进行注册（通过选择 **Tools»Advanced» .NET Assembly References**），LabVIEW 8.0 仍可能会提示您指定资源文件所在的位置。

Analyze VI 算法

在 LabVIEW 7.1 和更高版本中，Analyze VI 使用 BLAS/LAPACK 算法。这些 VI 现在能得出更精确的结果。在 LabVIEW 8.0 中，上述 VI 现被放置于 **Mathematics** 和 **Signal Processing** 选板中。

Append Signals Express VI

在 LabVIEW 7.x 中，如果 Append Signals Express VI 的 **Input Signal A** 为空或未连接，那么将单个信号或一组信号连接到 **Input Signal B** 时，**Appended Signals** 输出为空。在 LabVIEW 8.0 中，如果 **Input Signal A** 为空或未连接，那么将单个信号连接到 **Input Signal B** 时，Express VI 会返回 **Input Signal B**。如果将一组信号连接到 **Input**

Signal B, Express VI 会将该组信号依次拼接得到单个信号, 并将其作为结果输出。

比较函数

在 LabVIEW 7.x 和前期版本中, 使用比较 (Comparison) 函数比较 Variant 数据时, LabVIEW 将先比较两个 Variant 数据的长度, 然后逐位比较。LabVIEW 8.0 将先比较 Variant 数据的类型码 (type code), 该类型码记录了该 Variant 数据的实际类型信息, 然后再根据以上的类型信息, 对各类型所对应的数据加以比较。

Dot Product VI

在 LabVIEW 7.0 中, Dot Product VI 根据以下公式实现输入向量 X 和 Y 的点积计算:

$$X * Y = \sum_{i=0}^{n-1} x_i y_i$$

在 LabVIEW 7.1 及更高版本中, Dot Product VI 根据以下公式实现复数输入的点积计算:

$$X * Y = \sum_{i=0}^{n-1} x_i y_i^*$$

其中 y_i^* 是 y_i 的复共轭。

Easy Text Report VI (Mac OS 和 Linux)

LabVIEW 8.0 改变了 Easy Text Report VI 的接线器 (connector pane)。在 LabVIEW 8.0 中, 如果要使用已调用 Easy Text Report VI 作为子 VI 并在 LabVIEW 7.x 或前期版本中被保存的 VI, 必须右键单击程序框图中的子 VI 并从快捷菜单中选择 **Relink To SubVI**。

Format Into String 函数

在 LabVIEW 7.x 中, 将 %o 或 %x 指定格式的语法元素用于 Format Into String 函数时, 所输入的浮点数在进行字符串转换之前将被四舍五入为 32 位整数。

在 LabVIEW 8.0 中, 上述指定格式的语法元素使得该函数在进行字符串转换之前将浮点输入四舍五入为 64 位整数。

Join Numbers 函数

在 LabVIEW 7.x 和前期版本中，Join Numbers 函数会将 32 位整数输入强制转换为 16 位整数输入，并创建一个 32 位整数。在 LabVIEW 8.0 中，Join Numbers 函数可接受两个 32 位整数输入并创建一个 64 位整数。



注

如果在 LabVIEW 8.0 中打开 LabVIEW 7.x 所保存的 VI，LabVIEW 会将 Join Numbers 函数的 32 位整数输入强制转换为 16 位整数输入。

Number to String Conversion 函数

在 LabVIEW 7.x 中，将输入转换为字符串之前，Number to Hexadecimal String、Number to Octal String 和 Number to Decimal String 函数会将所输入浮点数四舍五入为 32 位整数。

在 LabVIEW 8.0 中，将输入转换为字符串之前，上述函数会把浮点输入四舍五入为 64 位整数。但如果在 LabVIEW 8.0 中打开由 LabVIEW 7.x 保存的 VI 时，LabVIEW 会将浮点输入四舍五入为 32 位整数后传递至以上函数，以确保兼容和功能正确。

Open VI Reference 函数

在 LabVIEW 7.x 中，如果 Open VI Reference 函数的 **vi path** 输入端是一个路径，并且内存中已经存在相同名称的 VI，那么即使内存中已有 VI 的路径与所指定路径不同，LabVIEW 也将返回一个指向内存中相应 VI 的引用。

在 LabVIEW 8.0 中，如果连接到 Open VI Reference 函数的 **vi path** 输入端是字符串，那么只有当 **vi path** 与所对应的终端内存中的某个 VI 的合法名称完全匹配时，LabVIEW 才会打开该 VI 引用。如果输入的 **vi path** 是路径，那么 LabVIEW 将在相应的终端内存中搜索具有相同路径的 VI。如果 LabVIEW 在内存中无法找到匹配路径的 VI，将会尝试从磁盘指定路径中加载 VI。如 LabVIEW 无法找到文件或该文件与内存中另一 VI 发生冲突，将会给出错误信息。关于合法文件名的详细信息见本文档的 LabVIEW 项目库一节。关于终端的详细信息见本文档的 LabVIEW 项目一节。

Quick Scale VI

在 LabVIEW 7.1 和前期版本中，如果 Quick Scale 1D VI 或 Quick Scale 2D VI 的 **X** 输入端是一个元素为零的数组，该 VI 所返回的 **max|X|** 将为 0，所返回的 $Y(i)=X(i)/\text{Max}|X|$ 或 $Yij=Xij/\text{Max}|X|$ 是元素为 NaN 的数组。在 LabVIEW 8.0 中，如果 Quick Scale VI 的 **X** 输入端为一个元素为零的数组，该 VI 所返回的 **max|X|** 将为 0，同时所返回的 $Y(i)=X(i)/\text{Max}|X|$ 或 $Yij=Xij/\text{Max}|X|$ 也是元素为零的数组。

Read Key VI

在 LabVIEW 7.x 和前期版本中，可以使用 Read Key VI 读取 Shift-JIS 编码的日语多字节字符串。同时必须将 1 或 <Shift-JIS> 连接到 **multibyte encoding** 输入端。在 LabVIEW 8.0 中，只要在操作系统语言环境

(locale) 中设置了相应编码方式，Read Key VI 便默认根据该设置读取多字节字符的编码字符串。

Scale VI

在 LabVIEW 7.1 和前期版本中，如果 Scale 1D VI 或 Scale 2D VI 的 **X** 输入端为一个元素为零的数组，该 VI 所返回的 **scale**、**offset** 和 **Y=(X-offset)/scale** 的值将分别为 0、0 和元素为 NaN 的数组。在 LabVIEW 8.0 中，如果 Scale VI 的 **X** 输入端为一个元素为零的数组，该 VI 所返回的 **scale**、**offset** 和 **Y=(X-offset)/scale** 的值将分别为 1、0 和元素为零的数组。

Semaphore VI

在 LabVIEW 7.x 中，Release Semaphore VI 和 Acquire Semaphore VI 在 **error in** 参数显示有错误时将不会执行。在 LabVIEW 8.0 中，即使 **error in** 参数显示有错误上述 VI 也会尝试执行。但如果在 LabVIEW 8.0 中打开 LabVIEW 7.x 的 VI，LabVIEW 会保持 LabVIEW 7.x 的原有功能。

SMTP Email VI

在 LabVIEW 7.x 和前期版本中，输入某特定字符串到 SMTP Email VI 的 **character set** 输入端即可确定一个字符集。而在 LabVIEW 8.0 中，SMTP Email VI 假定字符集信息位于系统字符集中。上述 VI 在发送电子邮件之前会将信息编码为 UTF-8 格式。SMTP Email VI 不再包括 **character set** 和 **translit** 参数。

Sort Complex Numbers VI

在 LabVIEW 7.x 和前期版本中，如果将 Sort Complex Numbers VI 的 **method** 输入端设置为 **Magnitude**，LabVIEW 不会改变模值相同的元素的先后顺序。在 LabVIEW 8.0 中，如果将 **method** 设置为 **Magnitude**，LabVIEW 处理模值相同的元素时会依次根据元素的实部和虚部进行排序。

Unit Vector VI

在 LabVIEW 7.x 和前期版本中，Unit Vector VI 可根据下面的公式计算输入向量的范数：

$$||X|| = \sqrt{x_0^2 + x_1^2 + \dots + x_{n-1}^2}$$

LabVIEW 8.0 中，Unit Vector VI 可根据下面的公式计算输入向量的范数：

$$||X|| = \left(|x_0|^y + |x_1|^y + \dots + |x_{n-1}|^y \right)^{\frac{1}{y}}$$

其中 X 为输入向量， $||X||$ 为范数， y 为范数类型。

User VI

在 labview\help、labview\project 或 labview\wizard 目录中的 VI 将分别出现在 **Help**、**Tools** 和 **File** 菜单中。在 LabVIEW 8.0 中，对这些 VI 的处理方式以及这些 VI 的工作模式与 LabVIEW 7.x 和前期版本不同，因为 LabVIEW 8.0 将在一个专用程序实例中打开上述 VI。

labview\vi.lib\Utility\allVIsInMemory.llb 中的 VIMemory Get VIs in Memory VI 可用于生成一个包括内存中所有程序实例中的所有用户 VI 的列表。labview\vi.lib\Utility\allVIsInMemory.llb 中的 Get User Application Reference VI 用于创建一个到当前程序实例的引用。关于程序实例的详细信息见本文档的运行程序实例一节。

不再支持的 VI 和函数

LabVIEW 8.0 不再支持以下 VI 和函数：

- LabVIEW 7.1 及更高版本中不再安装 Polynomial Real Zero Counter VI，现可使用 Polynomial Real Zeros Counter VI。
- 在 LabVIEW 7.1 及更高版本中不再安装 PPC VI，现可使用 TCP VI。
- LabVIEW 8.0 不再支持 QR Factorization VI，现可使用 QR Decomposition VI。
- LabVIEW 8.0 不再支持 Levenberg Marquardt VI 或 Nonlinear Lev-Mar Fit VI，现可使用 Nonlinear Curve Fit VI。
- 在 LabVIEW 8.0 中，**函数** 选板不再包括 VISA Status Description 函数，现可使用 Simple Error Handler 或 General Error Handler VI。
- LabVIEW 8.0 不再支持 Chi Square Distribution、F Distribution、Normal Distribution 和 T Distribution VI，现已分别替换为 Continuous CDF VI 的 Chi-Squared、F、Normal 和 Student t 实例。
- LabVIEW 8.0 不再支持 Inv Chi Square Distribution、Inv F Distribution、Inv Normal Distribution 和 Inv T Distribution VI，现已分别替换为 Continuous Inverse CDF VI 的 Use the Chi-Squared、F、Normal 和 Student t 实例。
- 在 LabVIEW 8.0 中，**函数** 选板不再包括 1D Linear Evaluation VI 和 2D Linear Evaluation VI，现可使用 Linear Evaluation VI。
- 在 LabVIEW 8.0 中，**函数** 选板不再包括 1D Polynomial Evaluation VI 和 2D Polynomial Evaluation VI，现可使用 Polynomial Evaluation VI。
- 在 LabVIEW 8.0 中，**函数** 选板中不再包括 1D Rectangular to Polar VI 和 1D Polar to Rectangular VI，现可使用 Re/Im To Polar 函数和 Polar To Re/Im 函数。
- 在 LabVIEW 8.0 中，**函数** 选板中不再包括 Harmonic Analyzer VI，现可使用 Harmonic Distortion Analyzer VI 测量得到 **THD** 或 **component levels** 输出，或者使用 SINAD Analyzer VI 测量得到 **SINAD** 或 **THD Plus Noise** 输出。

- 在LabVIEW 8.0中，**函数**选板不再包括Network Functions (avg) VI，现可使用 Frequency Response Function (Mag-Phase)、Frequency Response Function (Real-Im)、Cross Spectrum (Mag-Phase) 或 Cross Spectrum (Real-Im) VI。
- 在LabVIEW 8.0中，**函数**选板中不再包括Pulse Parameters VI，现可分别使用以下VI：其中Transition Measurements VI用于测量 **slew rate**、**duration**、**overshoot** 或 **preshoot** 输出，Pulse Measurements VI用于测量 **period**、**pulse duration** 或 **duty cycle** 输出，Amplitude and Levels VI用于测量 **amplitude**、**high state level** 或 **low state level** 输出。
- 在LabVIEW 8.0中，**函数**选板中不再包括Transfer Function VI，现可使用 Frequency Response Function (Mag-Phase) 或 Frequency Response Function (Real-Im) VI。
- 在LabVIEW 8.0中，**函数**选板不再包括NI DIAdem Report Wizard Express VI，现可使用NI DIAdem Report Express VI。关于NI DIAdem Report Express VI的详细信息见本文档的NI DIAdem Report Express VI一节。



注

使用NI DIAdem Report Express VI之前，必须安装DIAdem 9.1 Service Pack 2或更高版本。关于下载DIAdem试用版或购买DIAdem的详细信息，请登录National Instruments网站 ni.com 并输入信息代码 `exf5ty` 进行查询。

- **函数**选板不再包括VISA Resource Name和IVI Logical Name常量。现可使用VISA VI的 **VISA resource name** 输入端确定VISA资源名称，或使用相应的仪器驱动程序VI的相应输入端确定IVI逻辑名称。
- 在LabVIEW 8.0中，**函数**选板中不再包括error ring常量，现可使用32位有符号整数常量输入所需错误代码。
- **(Windows)** 在LabVIEW 8.0中，**函数**选板现已不包括原来LabVIEW 7.x **Sound** 选板中的Sound VI。请使用LabVIEW 8.0提供的相关Sound VI。LabVIEW 8.0也不再包括LabVIEW 7.x版本中的相关VI范例。
- **(Mac OS 和 Linux)** LabVIEW 8.0仍将包含LabVIEW 7.1的Sound VI。LabVIEW 8.0不再包括LabVIEW 7.x版本中的相关VI范例。

文件 I/O (File I/O) VI 和函数

在LabVIEW 8.0中，**函数**选板不再包括Read Characters From File VI，现可使用Read from Text File函数。

在LabVIEW 8.0中，**函数**选板不再包括Open/Create/Replace File VI，现可使用Open/Create/Replace File函数。以下函数实现了LabVIEW 7.x和前期版本中Open/Create/Replace File VI的部分功能。

- Get File Size 函数可用于确定文件大小。

- File Dialog Express VI 可用于指定供用户浏览选择文件的对话框中的起始路径、文件模式、默认文件名或路径名。
- Refnum to Path 函数可用于将一个文件引用转换为对应的文件路径。
- Write to Binary File 函数可用于创建平台独立的文本文件或其它类型的二进制文件，Read from Binary File 函数可读取对应的二进制文件。

在 LabVIEW 8.0 中，**函数**选板不再包括 Read File 和 Write File 函数，现可使用 Read from Binary File 和 Write to Binary File 函数。

在 LabVIEW 8.0 中，**函数**选板不再包括 Write Characters To File VI，现可使用 Write to Text File 函数。

在 LabVIEW 8.0 中，**函数**选板中不再包括 Access Rights 函数，现可使用 Get Permissions 和 Set Permissions 函数。

在 LabVIEW 8.0 中，**函数**选板中不再包括 EOF 函数，现可使用 Get File Size 和 Set File Size 函数。

在 LabVIEW 8.0 中，**函数**选板中不再包括 List Directory 函数，现可使用 List Folder 函数。

在 LabVIEW 8.0 中，**函数**选板中不再包括 Lock Range 函数，现可使用 Deny Access 函数。

如果打开一个用 LabVIEW 7.x 编写的 VI，并且该 VI 的程序框图中存在 New Directory 函数，LabVIEW 8.0 会将该函数替换为 Create Folder 函数。在 VI 运行时，如果 **path** 输入端所指定的文件夹不存在，Create Folder 函数将创建对应的目录结构，而不是像 New Directory 函数那样返回错误信息。

在 LabVIEW 8.0 中，**函数**选板中不再包括 Seek 函数，现可使用 Get File Position 和 Set File Position 函数。

在 LabVIEW 8.0 中，**函数**选板中不再包括 Type and Creator 函数，现可使用 Get Type and Creator 和 Set Type and Creator 函数。

在 LabVIEW 8.0 中，**函数**选板中不再包括 Volume Info 函数，现可使用 Get Volume Info 函数。

在 LabVIEW 8.0 中，**函数**选板不再包括 Open File 和 New File 函数，Read Lines From File VI 也不再出现在**函数**选板中，但为了保证兼容，LabVIEW 8.0 中仍将包含该 VI。

在 LabVIEW 8.0 中，**函数**选板不再包括 Read From I16 File、Read From SGL File、Write To I16 File 和 Write To SGL File VI。现可使用 Read from Binary File 和 Write to Binary File VI。

查找过时 VI

请按以下步骤查找在 LabVIEW 前期版本所创建的过时的 VI。

1. 将需要进行查找的 VI 加载至内存。
2. 选择 **Edit»Find and Replace** 显示 **Find** 对话框。
3. 单击 **Text** 按钮。
4. 在 **Application instance** 下拉菜单中，选择包含所查找 VI 的程序实例。关于程序实例的详细信息见本文档的运行程序实例一节。
5. 在 **Search Scope** 下拉菜单中选择 **All VIs in Application Instance**。
6. 勾选 **Hierarchy window** 复选框。
7. 单击 **Find** 按钮。

属性、方法和事件行为更改

LabVIEW 8.0 中包括以下属性、方法和事件更改。

应用程序的属性和方法

在 LabVIEW 8.0 中，部分应用程序属性和方法的行为取决于对应的程序实例。例如，**Application:All VIs in Memory** 属性的行为取决于所使用的程序实例。它将返回一个列表，包含了所有内存中与该属性对应的程序实例下的所有 VI。但是 **Application:Directory Path** 属性的行为并不取决于所使用的程序实例。该属性将返回包含应用程序的目录的绝对路径。该信息在所有程序实例中都将保持一致。

关于程序实例的详细信息见本文档的运行程序实例一节。

Front Panel:Open 方法

在 LabVIEW 7.1 中，LabVIEW 7.0 的 **Open FP** 方法已重命名为 **Old Open FP** 方法。LabVIEW 7.1 中另有一个不同的 **Open FP** 方法，该方法在前面板已经打开的情况下也不会返回错误信息。在 LabVIEW 8.0 中，LabVIEW 7.1 的 **Open FP** 方法已重命名为 **Front Panel:Open** 方法。如果 VI 中已使用 LabVIEW 7.0 的 **Old Open FP** 方法，请将该方法替换为 **Front Panel:Open** 方法。

Key Down 和 Key Repeat 事件

用于 VI 和控件的 **Key Down**、**Key Down?**、**Key Repeat** 和 **Key Repeat?** 事件的 **VKey** 数据栏现已具有区分的 **<Return>** 和 **<Enter>** 键值，其中 **Return** 和 **Enter** 键分别对应于主键盘和数字小键盘部分的回车键。在 LabVIEW 7.x 和前期版本中，**<Enter>** 键或 **<Return>** 键产生上述任一事件时，LabVIEW 将在 **VKey** 数据栏中返回 **<Enter>**。在 LabVIEW 8.0 中，按下 **<Enter>** 键或 **<Return>** 键产生上述任一事件时，LabVIEW 将在 **VKey** 数据栏中分别返回 **<Enter>** 或 **<Return>**。

(Mac OS) LabVIEW 8.0 现在仅支持快捷菜单的 <Control> 按键组合，而不再支持 <Command> 按键组合。如果使用事件结构实现这一操作，必须修改相应 VI。

列表框属性

在 LabVIEW 7.x 和前期版本中，如果将列表框 (listbox) 的 Top Row 设置为低于该列表框所有的最后一行的值，LabVIEW 会根据最后一行来决定显示结果。在 LabVIEW 8.0 中，该属性所接受并执行的行数值不再受到该列表框显示的行数的限制。

LabVIEW 8.0 不再支持单列列表框的 Double-Click 属性，现可使用 Get Double-Clicked Row 方法。

Owning VI 属性

在 LabVIEW 7.x 和前期版本中，Owning VI 属性返回一个对象所在 VI 的引用。该引用可将 VI 保留在内存中。在 LabVIEW 8.0 中，Owning VI 属性所返回的引用并不会将 VI 保留在内存中。如果对应的 VI 被从内存中移出，该引用将不再有效。使用 Open VI Reference 函数可获取内存中一个 VI 的引用，并且能将该 VI 保留在内存中，除非用户显式地关闭该引用。

文本属性

在 LabVIEW 7.x 和前期版本中，文本属性 (Text Property) 返回正常显示 (Normal Display) 的字符串。在 LabVIEW 8.0 中，文本属性所返回的字符串显示方式与前面板对象的显示方式相同。例如，如果使用密码显示 (Password Display) 的方式显示字符串控件，文本属性所返回的字符串的显示方式也是密码显示。

树形控件属性

在 LabVIEW 7.x 和前期版本中，Active Cell Properties:Cell Size:Height 和 Active Cell Properties:Cell Size:Width 属性将为树形控件 (TreeControl) 中的每行返回 17 个像素点。在 LabVIEW 8.0 中，Active Cell:Cell Size:Height 和 Active Cell:Cell Size:Width 属性将为树形控件中的每行返回 16 个像素点。

VI Strings 方法

在 LabVIEW 前期版本中使用 Export VI Strings 方法导出的字符串，可能不能被 LabVIEW 8.0 的 VI Strings:Import 方法正常导入。

不再支持的属性、方法和事件

LabVIEW 8.0 不再支持以下属性、方法和事件。

光标属性

LabVIEW 8.0 不再支持 Cursor Lock Style 属性，现可使用 Cursor Mode 属性。

列表框 (ListBox)、表格 (Table)、数字表格 (DigitalTable) 和树形控件 (TreeControl) 属性和事件

LabVIEW 8.0 不再支持多列列表框的 Cell Foreground Color 属性，现可使用 Active Cell:Cell Font:Color 属性。

LabVIEW 8.0 不再支持表格或数字表格的 Cell FG Color 属性，现可使用 Active Cell:Cell Font:Color 属性。

LabVIEW 8.0 不再支持树形控件的 Active Cell Properties:Foreground Color 属性，现可使用 Active Cell:Cell Font:Color 属性。

LabVIEW 8.0 不再支持树形控件类中的 Drag、Drag?、Drop 和 Drop? 事件。现可使用控件类中的 Drag Ended、Drag Enter、Drag Leave、Drag Over、Drag Source Update、Drag Starting、Drag Starting? 和 Drop 事件。

NamedNumeric 属性

LabVIEW 8.0 不再支持 Named Numeric Object 的 Named Numeric Colors、Named Numeric Colors:BG Color 或 Named Numeric Colors:Text Color 属性。现可分别使用 Text Colors、Text Colors:BG Color 和 Text Colors:Text Color 属性。

面板属性

LabVIEW 8.0 不再支持面板类 (Panel class) 的 Color 属性。如果在 LabVIEW 中使用该属性，属性仅应用于左上角的面板。现可使用面板类中的 Pane Color 属性。

子面板属性

在 LabVIEW 8.0 中，如使用子面板控件 (subpanel control)，可以用子 VI 的前面板设置子 VI 前面板在子面板控件中的显示位置，还可以设置是否显示它的滚动条。

LabVIEW 8.0 不再支持子面板控件的 X Scrollbar Visible 属性，现可使用面板的 Horizontal Scrollbar Visibility 属性。

LabVIEW 8.0 不再支持子面板控件的 Y Scrollbar Visible 属性，现可使用面板的 Vertical Scrollbar Visibility 属性。

LabVIEW 8.0 不再支持子面板控件的 Scale Panel 属性，现可使用面板的 Set Scaling Mode 方法。

VI 属性、方法和事件

LabVIEW 8.0 不再支持 Front Panel Window:Auto Center 属性，现可使用 Front Panel:Center 方法。

LabVIEW 8.0 不再支持 Front Panel Window:Size to Screen 属性，现可使用 Front Panel Window:State 属性。

LabVIEW 8.0 不再支持 VI 类的 Front Panel Window:Origin 属性。如果在 LabVIEW 中使用该属性，只有左上角的面板会应用该属性。现可使用面板类中的 Origin 属性。

LabVIEW 8.0 不再支持 VI 类的 Front Panel Window:Show Scroll Bars 属性。如果在 LabVIEW 中使用该属性，只有左上角的面板会应用该属性。现可使用面板类中的 Horizontal Scrollbar Visibility 和 Vertical Scrollbar Visibility 属性。

LabVIEW 8.0 不再支持 VI 类的 Get Front Panel Scaling Mode 或 Set Front Panel Scaling Mode 方法。如果在 LabVIEW 8.0 中使用这些方法，只有左上角的面板会应用它们。现可使用面板类中的 Get Scaling Mode 和 Set Scaling Mode 方法。

LabVIEW 8.0 不再支持 VI 类中的 Mouse Down、Mouse Down?、Mouse Enter、Mouse Leave、Mouse Move 或 Mouse Up 事件。现可分别使用面板类中的 Mouse Down、Mouse Down?、Mouse Enter、Mouse Leave、Mouse Move 和 Mouse Up 事件。

应用程序选项菜单标识符

LabVIEW 中已删除以下应用程序选项菜单标识符 (application item menu tag):

- APP_BUILD_STANDALONE_APP
- APP_DN_ASSEMBLY_REFS
- APP_EDIT_VI_LIBRARY
- APP_SAVE_WITH_OPTIONS
- APP_SHOW_CLIPBOARD
- APP_SRC_CODE_CTRL
- APP_SWITCH_EXEC_TARGET
- APP_UPDATE_VXI
- APP_VIEW_PRINTED_MANUALS

如果使用由 LabVIEW 前期版本保存的运行菜单 (.rtm) 文件，并且该文件包含一个已删除的标识符，那么在 LabVIEW 8.0 的 **Menu Editor** 对话框中保存该文件时，LabVIEW 会自动从 .rtm 文件中删除该标识符。已删除的应用程序选项标识符已保留为 LabVIEW 系统标识符，不能作为用户标识符进行使用。

对 HiQ 的支持

在 LabVIEW 8.0 中不再支持 HiQ。如果应用程序已使用 HiQ VI，可以考虑将其替换为 Mathematics 和 Signal Processing VI。关于使用 Mathematics 和 Signal Processing VI 的详细信息见 *LabVIEW Help*。

错误列表窗口

在 LabVIEW 7.x 和前期版本中，**错误列表 (Error list)** 窗口的 **VI List** 部分可显示内存中所有 VI 的错误。在 LabVIEW 8.0 中，**Error list** 窗口的 **Items with errors** 部分可显示内存中所有的错误，如 VI 和库等。如果存在两个或多个同名的条目，该部分会分别显示各个条目所属的特定程序实例。关于程序实例的详细信息见本文档的运行程序实例一节。

VI 字符串文件语法

通过选择 **Tools»Advanced»Import Strings** 选项或使用 VI Strings:Import 方法导入 VI 字符串文件 (VI String File) 时，LabVIEW 8.0 会搜索一组新的标识符：<GROUPER></GROUPER>。该组标识符指示了已被分组的前面板对象。因此 LabVIEW 8.0 无法导入由 LabVIEW 前期版本所保存的 VI 字符串文件。

在 LabVIEW 7.1 和前期版本中，列表框 (listbox) 的字符串位于专用数据 (private data) 的 <ITEMS> 部分。在 LabVIEW 8.0 中，列表框的字符串位于专用数据的 <STRINGS> 部分。同时在 LabVIEW 7.1 和前期版本中，列表框只能使用一种字体，该字体将在专用数据的 <LBLABEL> 部分列出。在 LabVIEW 8.0 中，列表框可使用多种字体，该字体在专用数据的 <CELL_FONTS> 部分列出。

在 LabVIEW 7.1 和前期版本中，多列列表框的字符串位于默认数据 (default data) 中。不过多列列表框的默认数据类型是整数或整数数组。在 LabVIEW 8.0 中，多列列表框的字符串位于专用数据中。

在 LabVIEW 7.1 和前期版本中，树形控件 (tree controls) 无法导出它包含的字符串及所用的字体。在 LabVIEW 8.0 中，树形控件可以导出所包含的字符串及所用的字体，所导出的格式与列表框或多列列表框的格式相同。

在 LabVIEW 8.0 生成的导出文件中，专用或默认数据的每一行不会包括超过两个标识符号。同时嵌套层数增加一层，LabVIEW 就会相应增加一级缩进。

请按以下步骤将 VI 字符串文件 (VI string file) 转换为 LabVIEW 8.0 格式：

1. 在 LabVIEW 前期版本中导入该 VI 字符串文件。
2. 保存 VI。
3. 在 LabVIEW 8.0 中加载该 VI。
4. 选择 **Tools»Advanced»Export Strings**，将 VI 字符串文件保存为 LabVIEW 8.0 格式。

将类型描述符数据与 LabVIEW 7.x 进行相互转换

在 LabVIEW 8.0 中，类型描述符 (Type Descriptor) 的保存格式已更改。LabVIEW 7.x 以 16 位的 I16 数组保存类型描述符，而 LabVIEW 8.0 以 32 位的形式来保存类型描述符。上述更改取消了类型描述符大小 64KB 的限制。

LabVIEW 8.0 所提供的机制允许您读取或生成 LabVIEW 7.x 的类型描述符。Flatten To String 函数包括有一个 **Convert 7.x Data** 快捷菜单选项。如果右键单击函数并选择该菜单选项，函数会为输入数据生成可供 LabVIEW 7.x 读取的类型描述符。选择 **Convert 7.x Data** 快捷菜单选项并连接 **data string** 输出端后，LabVIEW 8.0 将在函数上标出红色的 7.x 符号，表示该函数生成的数据格式可被 LabVIEW 7.x 识别。通过选择 **Convert 7.x Data** 快捷菜单选项并取消勾选标记，从而避免进行数据转换。

在 LabVIEW 8.0 中，如加载由 LabVIEW 7.x 或前期版本所保存的 VI，LabVIEW 8.0 会自动设置 Flatten To String 函数的 **Convert 7.x Data** 属性，并且该函数的使用方法与 LabVIEW 7.x 和前期版本相同。如果 VI 需使用 LabVIEW 8.0 的类型描述符格式，右键单击 Flatten To String 函数并从快捷菜单中选择 **Convert 7.x Data**，取消勾选标记。如 VI 无需使用包含由 LabVIEW 7.x 和前期版本所写的数据的文件，并且无需与由 LabVIEW 7.x 或前期版本所运行的 VI 进行发送或接收数据操作，那么可以使用 LabVIEW 8.0 类型描述符格式。LabVIEW 以后版本可能不再支持早期类型描述符格式。

将 NaN 字符串转换为整数类型 (Windows)

在 LabVIEW 7.x 中，将 NaN 字符串显式或隐式地转换为某整数类型时，其值将变为该整数数据类型的最小值。例如将 NaN 转换为 16 位有符号整数的结果为 -32768（16 位有符号整数能表示的最小值）。

在 LabVIEW 8.0 中，将 NaN 显式或隐式地转换为某整数类型时，其值将变为该整数数据类型的最大值。例如将 NaN 转换为 16 位有符号整数的结果为 32767（16 位有符号整数能表示的最大值）。

连接到条件结构的常量

在 LabVIEW 7.x 和前期版本中，如果将一个常量连接到条件结构 (Case Structure) 并将子 VI 置于没有执行的条件分支中，该子 VI 将一直被保留在内存中。例如，如果将 True 常量连接到条件结构，并将子 VI 置于条件结构的 False 条件分支中，LabVIEW 在加载调用 VI 的同时也会加载子 VI。LabVIEW 8.0 则将去除任何没有执行的代码，因此如果在 LabVIEW 8.0 中加载由 LabVIEW 前期版本所保存的 VI，并且 VI 包括一个连接到条件结构的常量，LabVIEW 会将常量转换为隐藏的控件 (hidden control) 以便与前期版本的行为保持一致。

延迟对操作系统消息的处理

在 LabVIEW 7.x 中，在运行用于处理 .NET 和 ActiveX 事件的回调 VI 的同时，LabVIEW 也可以处理操作系统消息。在 LabVIEW 8.0 中，LabVIEW 将延迟对操作系统消息的处理，直至回调 VI 执行结束或加载模态对话框时才处理系统信息。这样处理后，不仅回调 VI 可以不被中断地连续运行，同时也防止了 LabVIEW 在一个事件中激活另一事件而导致死锁状态发生。

在回调 VI 中无法对非模态对话框进行同步调用。必须从回调 VI 中异步调用非模态对话框，也就是利用 Run VI 方法来调用该对话框并将 FALSE 布尔常量连接至该方法的 **Wait Until Done** 输入端。

在 LabVIEW 7.x 中，在运行 DLL（动态链接库）函数的同时，LabVIEW 也可以处理操作系统消息。在 LabVIEW 8.0 中，LabVIEW 将延迟对操作系统消息的处理，直至结束调用 DLL 函数或者加载模态对话框时才处理系统信息。这样处理后，DLL 函数可以不被中断地连续运行，同时也防止了 LabVIEW 在运行 DLL 函数时调用同一个 DLL 而导致死锁状态的发生。

如果使用该默认行为，在运行 DLL 时将无法对非模态对话框进行同步调用。必须从 DLL 中异步调用非模态对话框，也就是利用 Run VI 方法来调用该对话框并将 FALSE 布尔常量连接至该方法的 **Wait Until Done** 输入端。

在所创建的 DLL 中可以选择是否延迟处理操作系统消息。查看 **My DLL Settings** 对话框的 **Advanced** 选项卡，并取消勾选 **Delay operating system messages in shared library** 复选框，即可以在 DLL 函数运行的同时处理操作系统消息。

资源管理器 (Mac OS)

LabVIEW 7.x 和前期版本提供的未正式编档的功能可帮助您读写 Macintosh 的资源文件。LabVIEW 8.0 不再支持该方法，因此无法直接从 VI 中读写 Macintosh 资源文件。

单按钮和双按钮对话框

在 LabVIEW 7.x 和前期版本中，无法通过编程来中止显示有单按钮对话框 (One Button Dialog) 或双按钮对话框 (Two Button Dialog) 的 VI。在 LabVIEW 8.0 中，利用 Abort VI 方法可以通过编程来中止显示有上述对话框的 VI。

属性和调用节点

如果在 LabVIEW 7.x 中创建有隐式连接到游标图例 (cursor legend) 的属性节点 (Property Node) 和调用节点 (Invoke Node)，在 LabVIEW 8.0 中打开该 VI 时 LabVIEW 将删除上述节点。

更新动态链接库

如果在 LabVIEW 7.x 或前期版本中创建一个链接到 labview.lib 的动态链接库 (DLL)，在 LabVIEW 8.0 中该动态链接库应链接至

labviewv.lib。关于在 LabVIEW 8.0 中使用动态链接库的详细信息见本文档的在各个版本的 LabVIEW 中使用动态链接库一节。

打印的页边距值

在 LabVIEW 7.x 和前期版本中，**Options** 对话框中 **Printing** 选项卡的 **Margins** 选项页边距值的单位是厘米，在 LabVIEW 8.0 中，**Margins** 选项使用毫米单位。

升级 LabVIEW 6.x

从 LabVIEW 6.x 升级至 LabVIEW 8.0 时，可能会遇到以下兼容性问题。关于可能碰到的升级问题的详细信息见本文档的升级 LabVIEW 7.x 一节。

关于各个版本的新功能和更改的详细信息，请登录 ni.com/manuals 并查看 LabVIEW 6.x 和 8.0 之间各个版本的《*LabVIEW 升级说明*》。

波形数据类型更改

在 LabVIEW 7.0 中，波形数据类型 (Waveform Data Type) 的 **td** 成员现已使用时间标记数据类型 (time stamp data type) 而不是双精度或浮点数字。在 LabVIEW 6.x 中，如果将数据以波形数据类型的格式保存到文件中而未包括数据类型的信息，在 LabVIEW 7.x 和更高版本中重新打开该数据时可能会发生错误。

在 LabVIEW 7.x 和更高版本中，Read Waveform from File VI 可将文件中的旧波形数据类型格式转换为新波形数据类型格式。该 VI 会显示一个对话框提示您进行转换确认。在仅有 LabVIEW 运行引擎 (Run-Time Engine) 运行的情况下，Read Waveform from File VI 无法执行该转换并将返回错误信息。关于将 LabVIEW 6.x 的波形数据转换至 LabVIEW 7.x 和更高版本的详细信息，请登录 National Instruments 网站 ni.com/info 并输入信息代码 `exd9zq` 进行查询。

Serial Compatibility VI

在 LabVIEW 7.x 和更高版本中，**函数** 选板中不再包括 Serial Compatibility VI，可使用 VISA VI 和函数创建用于 VXI 设备通信的 VI。

在 LabVIEW 7.x 和更高版本中，LabVIEW 不再通过 `serpdrv` 驱动程序与操作系统的串口驱动器进行通信，而提供了基于 VISA 的 VI 来实现对应的功能。开发新应用程序时，请利用 VISA 和 Serial VI 及函数控制串行设备。如果由 LabVIEW 前期版本创建的 VI 包括 Serial VI，这些 VI 仍可以继续 LabVIEW 7.1 及更高版本中运行。

如果重新配置端口号和端口的映射，必须给各个端口确定一个映射。使用 `labview\vi.lib\Instr\_sersup.lib` 中的 Set Serial Alias Ports VI 可确定串口映射关系。将一个字符串数组连接到 VI 的 **VISA Aliases** 输入端，并且注明输入数组所使用的端口名称。数组中各个元素必须与某个端口对应。比如，如果配置端口 0 映射 VISA alias MySerialPort，输入

MySerialPort 作为 **VISA Aliases** 输入数组的第一个元素。调用 VISA Configure Serial Port VI 之前，必须先调用 set serial alias ports VI。

关于使用 VISA VI 和函数控制串行设备的范例信息见 labview\examples\instr\smplsrl.llb。

循环结构中的默认数据

在 LabVIEW 6.0 和前期版本中，如果 For 循环未执行，会产生未定义数据。在 LabVIEW 6.1 和更高版本中，如果将 0 连接到 For 循环的总数接线端或将空数组作为输入连接到 For 循环并且启用自动索引，For 循环将产生默认值。该循环将不会执行，任何禁用自动索引的输出隧道 (tunnel) 将返回该隧道数据类型的默认值。

远程前面板许可证

LabVIEW 完整版开发系统和 Application Builder(应用程序生成器) 包括一个远程前面板许可证 (Remote Front Panel License)，可允许一个客户端远程查看和控制前面板。LabVIEW 专业版开发系统包括一个远程前面板许可证，可允许五个客户端远程查看和控制前面板。

如需支持更多客户端，请升级远程前面板许可证。

多线程分配

相对于 LabVIEW 7.1 的前期版本，LabVIEW 7.1 和更高版本会分配更多的线程来执行 VI。基于这项改动，如所调用的 DLL 实际不支持可重入 (reentrant)，但对应的调用库函数节点 (Call Library Function Node) 标为可重入，多线程运行时可能会出现这个问题。关于调用库函数节点和可重入的详细信息见 *LabVIEW Help*。

labview\vi.lib\Utility\sysinfo.llb 中的线程配置 (threadconfig) VI 可用于配置 LabVIEW 分配线程的方式。也可以通过选择 **File»VI Properties**，并从 **Category** 下拉列表中选择 **Execution**，取消勾选 **Reentrant execution** 复选框，禁用 VI 的重入功能。

关于多线程分配的详细信息见 *LabVIEW Help*。

仪器驱动程序

LabVIEW 7.x 和更高版本中的 LabVIEW 中不再包括带有仪器驱动程序的 LabVIEW 仪器驱动库 CD 光盘，可通过 National Instruments 仪器驱动程序网站 ni.com/idnet 下载仪器驱动。National Instruments 设备驱动程序 CD 光盘包括 NI-DAQ、NI-VISA 和其它 National Instruments 驱动程序。

单位和换算因数

在 LabVIEW 7.x 和更高版本中，使用 Compound Arithmetic 函数后，无需再利用 Convert Unit 函数删除多余单位。

LabVIEW 7.1 和更高版本中的单位换算因数 (unit conversion factor) 更符合美国国家标准技术研究所 (NIST) 所发表的 *国际单位系统使用指南* (SI) 的指导方针。同时卡路里 (calorie) 单位和马力 (horse power) 现已分别更名为卡路里 (热量) (calorie (thermal)) 和马力 (电气) (horsepower (electric))。上述单位的缩写无任何改变。下表为 LabVIEW 6.1 和 8.0 中单位换算因数改动的详细信息。

表 1 单位换算因数

单位	6.1 定义	8.0 定义
天文单位 (astronomical unit) (AU)	149,498,845,000 m	149,597,900,000 m
英国热量单位 (British Thermal Unit) (mean)	1055.79 J	1055.87 J
电子伏 (electron volt) (eV)	1.602e-19 J	1.60217642e-19 J
英尺 - 烛光 (foot-candle)	10.764 lx	10.7639 lx
马力和马力 (电气) (horse power versus horsepower (electric))	745.7 W	746 W 新转换为精确值。
英国标准加仑 (imperial gallon)	4.54596 l	4.54609 l
光年 (light year)	9.4605 Pm	9.46073 Pm
磅力 (pound force)	4.448 N	4.448222 N
杆 (rod)	502.92 cm	5.029210 m
slug	14.594 kg	14.59390 kg
原子质量 (unified atomic mass) (u)	1.66057e-27 kg	1.66053873e-27 kg

Defer Panel Updates 属性

在 LabVIEW 6.1 和前期版本中，只有当 Defer Panel Updates 属性为 FALSE 时，LabVIEW 才会更新所有前面板对象的待处理修改。在 LabVIEW 8.0 中，将该属性设置为 TRUE 时，LabVIEW 会更新所有前面板对象的待处理修改，并推迟所有前面板更新的新请求。某些情况下，上述改动可能引起 LabVIEW 多更新一次前面板上发生变化的控件。

数值控件的数据范围

在 LabVIEW 6.1 和前期版本中，一些数值控件 (Numeric Control) 的默认最小值为 0.00、默认最大值为 0.00 以及默认增量值为 0.00，并且超出范围的操作为 **Ignore**。在 LabVIEW 8.0 中，上述数值控件将根据对应的数据类型来确定默认的数值范围。

强制转换点和自定义类型

在 LabVIEW 6.1 和更高版本中，连线包括自定义类型 (Type Definitions) 的信息，因而在程序框图中可能存在更多的强制转换点 (Coercion Dots)。将自定义类型连接到 VI 或函数接线端时，如果该 VI 或函数接线端不是该自定义类型接线端，会出现强制转换点。如将自定义类型的输出接线端连接到不匹配的显示控件时，也会出现强制转换点。强制转换点说明在 VI 中使用了不一致的自定义类型。这种情况下，强制转换点不会影响运行性能。

关于使用 Flatten To String 函数转换自定义类型的信息见 *LabVIEW Help*。

文件对话框按钮标签

在 LabVIEW 6.1 和前期版本的 File Dialog 函数所显示的文件对话框中，如果用户输入新文件名，该对话框包括有 **Save** 按钮，否则按钮标签 (Button Label) 为 **Open**。在 LabVIEW 8.0 中，在所有情况下 File Dialog Express VI 所显示的文件对话框按钮标签均为 **OK**，但允许用户进行修改。通过 File Dialog Express VI 的 **button label** 输入端可修改该按钮标签。如在现有 VI 中使用 File Dialog Express VI，请查看该 VI 的行为以确保 **OK** 的默认标签可与该 VI 的功能相匹配。

Control Online Help 函数

Control Online Help 函数的 **Path to the help file** 输入端为必须连接。可将已编译的帮助文件名 (.chm 或 .hlp) 或帮助文件的完整路径连接到该输入端。如只连接已编译的帮助文件名，LabVIEW 将在 labview\help 目录中查找该文件。

Run VI 方法

在 LabVIEW 7.1 中，如果将 Run VI 方法的 **Auto Dispose Ref** 输入端设置为 TRUE，VI 停止运行后 LabVIEW 将自动清除该引用。在 LabVIEW 8.0 中，如果该方法返回错误，LabVIEW 也可以马上清除该引用。如部分程序框图代码依赖于该引用，该 VI 在运行时可能无法正常执行。

加载时显示前面板

在 LabVIEW 8.0 中，如将 VI 配置为在 LabVIEW 加载时显示 VI 前面板，但通过 VI Server 加载该 VI，LabVIEW 不会显示前面板。必须使用 Front Panel:Open 方法来编程显示该前面板。

Open VI Reference 函数

在 LabVIEW 6.1 和前期版本中，如 Open VI Reference 函数的 **options** 参数未连接任何值，并且内存中未加载某 VI 模板，LabVIEW 会根据模板创建一个新 VI。如内存已加载该 VI 模板，LabVIEW 将返回一个该模板的引用。在 LabVIEW 7.0 和更高版本中，如果内存中已加载某模板，但未将 **options** 参数设置为 0x02，使用 Open VI Reference 函数创建该模板的引用将得到一个错误返回。

指数表示法

LabVIEW 6.0 和前期版本的公式节点 (Formula Node) 中使用 ^ 运算符表示指数操作。LabVIEW 6.1 和更高版本使用 ** 运算符表示指数运算，如 $x**y$ 。^ 运算符表示按位异或 (XOR) 操作。

IVI 配置文件

IVI 配置文件 (IVI Configuration Store File) 格式要求所有名称区分大小写。如果在应用程序中使用逻辑名称、驱动 (Driver Session) 名称或虚拟名称，请确保所使用名称可完全匹配 IVI 配置文件所定义的名称。

技术支持表格

在 LabVIEW 7.x 和更高版本中，LabVIEW 安装程序不安装 techsup.llb。关于解决安装、配置和应用程序问题等信息，请登录 National Instruments 网站 ni.com/support。

升级 LabVIEW 5.x

从 LabVIEW 5.x 升级至 LabVIEW 8.0 时，可能会遇到以下兼容性问题。关于可能碰到的升级问题的信息见本文档的升级 LabVIEW 6.x 和升级 LabVIEW 7.x 各节。

关于各个版本的新功能和改进的详细信息，请登录 ni.com/manuals 并查看 LabVIEW 5.x 和 8.0 之间各个版本的《LabVIEW 升级说明》以及 LabVIEW 5.1 Addendum 等信息。

转换数据记录文件

LabVIEW 8.0 会检查数据记录文件 (Datalog File) 的类型，确定是否需要转换。如果数据记录文件版本低于 LabVIEW 6.0，或者文件包含波形数据类型并且版本低于 LabVIEW 7.1，LabVIEW 8.0 将转换该文件以便读取和添加记录。在其它情况中，读取数据记录文件或者向数据记录文件添加记录均不会进行文件转换。

打开由 LabVIEW 前期版本创建的数据记录文件时，LabVIEW 8.0 会提示您将该文件转换为 LabVIEW 8.0 格式。如果选择进行转换，LabVIEW 会使用新格式的数据文件替换该数据记录文件；如果选择不转换，LabVIEW 8.0 将不会打开该文件并返回错误信息。



注

如仍需在 LabVIEW 6.1 或前期版本中使用旧数据文件，请在转换前备份该数据记录文件。LabVIEW 6.1 或前期版本无法对已转换的数据记录文件进行恢复或读取操作。

如需在打开数据记录文件时进行自动转换，请将下列行添加至 LabVIEW 预置文件 (preferences file):

```
silentDatalogConvert=True
```

(Mac OS) 添加以下行:

```
silentDataLogConvert:True
```

(Linux) 添加以下行:

```
labview.silentDataLogConvert:True
```

如不希望在打开数据记录文件时进行自动转换, 请将上面这些值设置为 False。

连接到 VI Server

由于 LabVIEW 5.x 应用程序无法识别 LabVIEW 8.0 VI Server 部分协议, LabVIEW 8.0 客户端将无法连接到 LabVIEW 5.x 应用程序的 VI Server。但 LabVIEW 5.x 客户端可以连接至 LabVIEW 8.0 应用程序的 VI Server。

UDP 函数

在 LabVIEW 6.x 和更高版本中, **函数**选板不再包括 UDP VI, 但为了保证兼容性, 这些 VI 仍被保存在 labview\vi.lib_oldvers_oldvers.lib 中。现可使用内置 UDP 函数以实现网络通信。

升级 LabVIEW 4.x

从 LabVIEW 4.x 升级至 LabVIEW 8.0 时, 可能遇到以下兼容性问题。关于可能碰到的升级问题的信息见本文档的 *升级 LabVIEW 5.x*、*升级 LabVIEW 6.x* 和 *升级 LabVIEW 7.x* 各节。

关于各个版本的新功能和改进的详细信息, 请登录 ni.com/manuals 并查看 LabVIEW 4.x 和 8.0 之间各个版本的《*LabVIEW 升级说明*》以及 *LabVIEW 5.1 Addendum* 等信息。

与 LabVIEW 4.x 之间的布尔数据转换

在 LabVIEW 4.x 和 LabVIEW 5.x 中, LabVIEW 已经改变布尔 (Boolean) 数据的保存格式。LabVIEW 4.x 保存的布尔数据为 2 字节, 只有当该数据为数组时, LabVIEW 4.x 才会将各个布尔元素保存为 1 位。无论该数据是否为数组, LabVIEW 8.0 都会将布尔值保存为 1 字节, 上述改动可允许更多程序框图函数支持布尔值数组, 并可使该数组与数字数组的行为保持一致。LabVIEW 8.0 布尔数据格式将影响代码接口节点 (Code Interface Nodes (CINs)) 的数据操作, 但 LabVIEW 8.0 也可兼容现有 CINs。

如果在 LabVIEW 4.x 中将包括一个或多个布尔值的二进制数据写入文件, 同时在 LabVIEW 8.0 中写入相同的数据, 两者的格式并不会相同。LabVIEW 8.0 所提供的机制可允许您读取 LabVIEW 4.x 所写的二进制数据或者写入二进制数据以供 LabVIEW 4.x 读取。Type Cast、Flatten To String 和 Unflatten From String 函数都已包括 **Convert 4.x Data** 快捷菜单选项。如右键单击函数并选择该菜单选项, 函数会将数据作为供 LabVIEW 4.x 使用的二进制数据进行操作。如需产生用于 LabVIEW 4.x 格式的数据, 请使用 Write to Binary File、Flatten To String 或 Type Cast

函数。在读取 LabVIEW 4.x 格式的数据时，请使用 Read from Binary File、Unflatten From String 或 Type Cast 函数。选择 **Convert 4.x Data** 快捷菜单选项后，LabVIEW 8.0 将在函数上标出红色的 4.x，用于表示该函数的数据格式可与 LabVIEW 4.x 相互转换。通过选择 **Convert 4.x Data** 快捷菜单选项并取消勾选标记，从而避免进行数据转换。

如多个文件都包括布尔值，可以创建一个 VI 以打开上述文件并将数据保存到 LabVIEW 8.0 的新文件格式中。

在 LabVIEW 8.0 中，如加载由 LabVIEW 4.x 和前期版本所保存的 VI，LabVIEW 8.0 会自动设置 Write to Binary File、Read from Binary File、Type Cast、Flatten To String 和 Unflatten From String 函数的 **Convert 4.x Data** 属性，这些函数可跟以前一样进行操作。如果 VI 需使用 LabVIEW 8.0 布尔数据格式，右键单击该函数并从快捷菜单中选择 **Convert 4.x Data**，取消勾选标记。如 VI 无需使用包含由 LabVIEW 4.x 和前期版本所写的布尔数据的文件，并且无需与由 LabVIEW 4.x 或前期版本所运行的 VI 进行发送或接收数据操作，可以使用 LabVIEW 8.0 布尔数据格式。LabVIEW 以后版本可能不再支持早期布尔数据格式。

VI Control VIs

在 LabVIEW 5.x 和更高版本中，**函数**选板不再包括 VI Control VIs，但该 VI 将作为可兼容 VI 保存在 labview\vi.lib\utility\vict1.llb 中。对于 VI Control VIs，现可使用 VI Server 函数、Open VI Reference、Call By Reference Node、Property Node 和 Invoke Node。

在 LabVIEW 8.0 中 VI Control VIs 所返回的部分错误代码将会发生变化。LabVIEW 前期版本中，VI Control VIs 返回的错误代码为 7 和 1000。LabVIEW 8.0 中 VI Control VIs 返回的代码为 1004 和 1003。如果 LabVIEW 4.x 所创建的 VI 需检查错误代码 7 和 1000，必须修改该 VI 以便在 LabVIEW 8.0 中运行。

DDE VI (Windows)

在 LabVIEW 5.x 和更高版本中，**函数**选板不再包括 DDE VI，但为保证兼容性，该 VI 仍被保存在 labview\vi.lib\platform\dde.llb。

升级 LabVIEW 3.x 和前期版本

关于使用 VI 转换工具包从 LabVIEW 3.x 或前期版本进行升级的详细信息，请登录 National Instruments 网站 ni.com。关于可能碰到的升级问题的信息见本文档的升级 LabVIEW 4.x、升级 LabVIEW 5.x、升级 LabVIEW 6.x 和升级 LabVIEW 7.x 各节。

关于各个版本的新功能和改进的详细信息，请登录 ni.com/manuals 并查看 LabVIEW 3.x 和 8.0 之间各个版本的《*LabVIEW 升级说明*》以及 *LabVIEW 5.1 Addendum* 等信息。

LabVIEW 8.0 功能和更改

关于 LabVIEW 编程理论、编程分步指导以及 LabVIEW 8.0 功能的详细信息见 *LabVIEW Help*。请在菜单中选择 **Help»Search the LabVIEW Help** 打开 *LabVIEW Help*。

关于 LabVIEW 8.0 功能和改动的详细信息见 `labview` 目录中的 `readme.html` 文件。

激活 LabVIEW 许可证

(Windows) LabVIEW 依赖于许可证激活机制，运行 LabVIEW 之前，必须使用合法的 LabVIEW 许可证进行激活。安装程序软件包中的序列号可以用来激活 LabVIEW。请使用以下任一方式激活 LabVIEW 许可证：

- 在安装过程中，输入序列号并选择完成后运行**激活向导 (Activation Wizard)**。
- 以试用版模式 (evaluation mode) 加载 LabVIEW 时，在**启动激活 (Activation Startup)** 对话框中选择**激活 (Activate)**。
- 以试用版模式运行 LabVIEW 后，选择 **Help»Activate LabVIEW**。必须重启 LabVIEW，许可证激活才会生效。

关于 LabVIEW 许可证激活的详细信息见 *LabVIEW 8.0 发行说明*。

启动 LabVIEW

运行 LabVIEW 时会出现 **Getting Started** 窗口，该窗口可用来创建新 VI 和项目 (projects)、选择最近打开的 LabVIEW 文件、查找范例以及查看 *LabVIEW Help* 等。该窗口也可用于查看各种信息和资源，了解 LabVIEW，如用户手册、帮助主题以及 National Instruments 网站的各种资源等。

打开现有文件或创建新文件后 **Getting Started** 窗口就会消失。通过选择 **View»Getting Started Window** 可以显示该窗口。关闭所有已打开的前面板和程序框图后也会重新显示 **Getting Started** 窗口。

登录或注销 LabVIEW

如果在网络中已经配置了一个安全域，可以登录 LabVIEW 开发环境。当与包括权限控制列表的远程 VI Server 通信时，通过登录信息可以识别 LabVIEW。NI Security:Login 和 NI Security:Logout 方法和 NI Security User Change 事件可以分别用于通过编程在程序框图中登录或注销 LabVIEW 以及监控用户帐号变动。

域帐号管理器

选择 **Tools»Security»Domain Account Manager** 运行域帐号管理器 (Domain Account Manager)，可以创建或删除域、实施域策略以及在本地或通过远程管理域中的用户或组帐号。在管理远程域时该管理器特别有用，因为用户可以远程控制用于管理域的远程主机。由于使用的是内置的数据编码和数据集成，该认证协议将更加安全。

新建对话框的改进

单击 **Getting Started** 窗口的 **New** 链接可显示**新建 (New)** 对话框。**New** 对话框可用于创建新 VI、项目 (projects)、项目库 (project libraries)、自定义控件 (custom controls)、共享变量 (shared variables) 等。通过勾选 **Add to project** 复选框可将该新建项加入到某个已打开的项目中。

在 LabVIEW 7.x 和前期版本中，选择 **Small dialog** 或 **Large dialog** 按钮，可缩小或扩大 **New** 对话框窗口大小。LabVIEW 8.0 并不包括 **Small dialog** 或 **Large dialog** 按钮，但调整 **New** 对话框大小后，LabVIEW 在下次打开该对话框时可保持同样大小。

在 LabVIEW 7.x 和前期版本中，**New** 对话框可预览所选择 VI 的前面板和程序框图。在 LabVIEW 8.0 中，**New** 对话框的 **Description** 部分包括显示程序框图预览。

Create New 列表不再像 LabVIEW 7.x 及前期版本那样包括 VI 模板、全局变量模板和控件模板选项。在 VI 或全局变量的前面板或程序框图中，选择 **File»Save As** 可显示文件对话框，在 **Save as type** 下拉菜单中选择 **Templates VIs**，可将条目保存为模板。在自定义控件的前面板中，选择 **File»Save As** 可显示文件对话框，在 **Save as type** 下拉菜单中选择 **Template Controls**，可将控件保存为模板。

RT、FPGA 和 PDA 终端

要对 RT、FPGA 或 PDA 终端进行操作，必须创建一个 LabVIEW 项目。关于在 LabVIEW RT、FPGA 和 PDA 模块中使用项目的详细信息见特定模块的相关文档。

切换执行终端

在 LabVIEW 7.x 和前期版本中，必须在 **LabVIEW** 对话框的 **Execution Target** 下拉菜单中选择一个终端，才可以确定用于开发 VI 的相应终端。如

果在开发某个 VI 时需切换至其它执行终端，可单击 **Operate»Switch Execution Target** 并选择所需终端。在 LabVIEW 8.0 中，利用项目开发环境可以对不同的执行终端进行操作。在 **Project Explorer** 窗口，右键单击项目根目录并从快捷菜单中选择 **New»Targets and Devices**，打开 **Add Targets and Devices** 对话框。如果该项目中的终端可支持其它终端，也可以右键单击该终端并从快捷菜单中选择 **New»Targets and Devices**，从而在现有终端中添加其它终端。在终端中可以添加和开发 VI，然后通过切换位于不同终端下的 VI，就可以在相应执行终端间进行切换。关于在 LabVIEW RT、FPGA 和 PDA 模块中切换执行终端的详细信息见相应模块的相关文档。

LabVIEW 项目

项目 (project) 可用于归档 LabVIEW 文件和非 LabVIEW 文件、创建程序生成规范以及在终端中部署或下载文件。终端是运行 VI 的设备或机器。保存一个项目时，LabVIEW 会创建一个项目文件 (.lvproj)，其中包括项目中文件的引用、配置、程序生成以及部署信息等。

必须使用项目来创建可独立执行的应用程序和动态链接库。同时也必须使用项目来对 RT、FPGA 或 PDA 终端进行操作。关于在 LabVIEW 实时 (RT)、FPGA 和 PDA 模块中使用项目的详细信息见特定模块的相关文档。

项目类型的 LabVIEW 即插即用仪器驱动程序可利用 LabVIEW 8.0 的项目和项目库功能。可采用与早期 LabVIEW 即插即用驱动程序同样的方式使用项目类型的驱动程序。

项目浏览器窗口

项目浏览器 (Project Explorer) 窗口用于创建和编辑 LabVIEW 项目。选择 **File»New Project** 以显示 **Project Explorer** 窗口。也可以选择 **Project»New Project** 或 **File»New**，然后在 **New** 对话框中选择 **Empty Project** 显示 **Project Explorer** 窗口。

默认情况下，**Project Explorer** 窗口包括以下条目：

- **项目根 (Project root)** — 包含 **Project Explorer** 窗口中的所有其它条目。项目根的标签包括该项目的文件名。
 - **我的电脑 (My Computer)** — 表示可作为项目中的一个终端来使用的本地计算机。
 - **依赖性 (Dependencies)** — 包括该终端中 VI 所必须的其它资源。
 - **程序生成规范 (Build Specification)** — 包括发布源代码的配置以及 LabVIEW 工具包和模块所支持的发布形式的配置。如果已经安装 LabVIEW 专业版开发系统或 Application Builder，可以使用 **程序生成规范** 来配置独立应用程序 (EXE)、动态链接库 (DLL) 和压缩文件。**(Windows)** 也可以使用 **程序生成规范** 配置安装程序 (installer) 的生成。

在项目中添加其它终端时，LabVIEW 会在 **Project Explorer** 窗口中创建新的条目来表示终端。各个终端也包括**依赖性**和**程序生成规范**，可以添加文件至各个终端。

从 **Project Explorer** 窗口中可以将 VI 拖放至其它已打开 VI 的程序框图中，并将其作为子 VI 使用。

在项目中使用已有文件

在项目中可以添加一组已有文件。请按以下任一方式将已有文件添加到项目中：

- 如果已有文件是独立应用程序的一部分，选择 **Tools»Convert Build Script** 将应用程序转换为一个项目。关于导入程序生成脚本的详细信息见本文档的 **Application Builder** 前期版本更改一节。
- 如果已有文件不是独立应用程序的一部分，可以创建一个新项目并添加文件至 **Project Explorer** 中。关于将文件添加至项目的详细信息见本文档的在项目中添加条目一节。

项目浏览器窗口工具栏

在 LabVIEW 项目中进行操作，可使用 **Standard**、**Project**、**Build Specifications** 和 **Source Control** 工具栏按钮。工具栏位于 **Project Explorer** 窗口顶端。可能需要调整 **Project Explorer** 窗口大小才可以查看所有工具栏。

通过单击 **View»Toolbars** 并选择相应的工具栏，可以更改窗口中所显示的 工具栏。也可以右键单击工具栏中的空白区域并选择需隐藏或显示的工具栏。

在项目中添加条目

使用 **Project Explorer** 窗口可将 LabVIEW 文件（如 VI 和 LLB）以及非 LabVIEW 文件（如文本文件和电子表格）添加到 LabVIEW 项目的终端中。也可以为项目中的条目创建一个组织结构。

通过以下方式可以将新条目添加到项目的终端中：

- 右键单击终端或终端的文件夹，并从快捷菜单中选择 **Add File** 可从文件对话框中选择需添加的文件。也可以选择终端，单击 **Project»Add To Project»Add File** 并从文件对话框中选择需添加的文件。
- 右键单击终端，并从快捷菜单中选择 **New»VI** 可添加新的空白 VI。也可以选择 **File»New VI** 或 **Project»Add To Project»New VI** 添加新的空白 VI。
- 选择前面板或程序框图右上角的 VI 图标并将其拖放到终端中。
- **(Windows 和 Mac OS)** 从磁盘选择一个条目或目录并将其拖放到终端中。

也可以从 **New** 对话框添加新 LabVIEW 文件到项目中。选择 **File»New** 或 **Project»Add To Project»New** 显示 **New** 对话框。在 **New** 对话框中，选择需添加的条目并勾选 **Add to project** 复选框。如果同时打开多个项目，可从 **Projects** 列表中选择需添加新条目的项目。

在项目中添加文件夹

Project Explorer 窗口可把磁盘中的目录作为文件夹添加至项目中。右键单击终端或终端中的文件夹，并从快捷菜单中选择 **Add Folder**，以添加磁盘目录中的文件。选择磁盘上的目录可在项目中创建表示整个目录内容的条目，其中包括文件和子文件夹的内容。LabVIEW 将使用与磁盘上目录相同的名称在项目中创建一个新虚拟文件夹 (virtual folder)。

也可以选择终端并单击 **Project»Add To Project»Add Folder**，从目录中添加文件。



注

将磁盘上的目录添加至项目后，如果对磁盘上的目录进行修改，LabVIEW 并不会自动更新项目中所对应的文件夹。如果对磁盘上的子目录结构进行修改，LabVIEW 也不会自动更新项目中所对应的子文件夹。如希望项目中的文件夹与磁盘上的目录相匹配，必须手动更新文件夹。

也可以在项目中创建新文件夹用于条目管理。右键单击终端，并从快捷菜单中选择 **New»Folder** 可将新文件夹添加到终端中。如需创建新子文件夹，右键单击现有文件夹并从快捷菜单中选择 **New»Folder**。

右键单击项目根，并从快捷菜单中选择 **New»Folder** 可添加新文件夹以便管理项目中的终端。

在项目中添加 LLB

可将 LLB 作为文件夹或文件添加至项目。如果将 LLB 作为文件夹进行添加，LabVIEW 会将 LLB 文件名作为该文件夹的名称并将 LLB 中的 VI 作为新文件夹中的条目添加至项目中。右键单击终端或终端中的文件夹，从快捷菜单中选择 **Add Folder**，并通过文件对话框查找需添加的 LLB。在 **Project Explorer** 的文件夹中添加或删除条目并不会影响磁盘上的 .llb 文件。选择 **Tools»LLB Manager** 可编辑磁盘上的 LLB。

也可将 LLB 作为文件添加至项目。如果将 LLB 作为文件进行添加，**Project Explorer** 窗口中并不会出现 LLB 所包括的 VI。右键单击终端或终端中的文件夹，从快捷菜单中选择 **Add File**，并通过文件对话框查找需添加的 LLB。选择右侧标有单个句点和打开文件夹图标项的 **VI**，并单击 **Select** 按钮。

在项目中删除条目

Project Explorer 窗口可用于在 LabVIEW 项目中删除条目。请按以下方式删除条目：

- 右键单击需删除的条目，并从快捷菜单中选择 **Remove**。
- 选择需删除的条目并按 <Delete> 键。

- 选择需删除的条目并单击 **Standard** 工具栏中的 **Delete** 按钮。



注

在项目中所删除的条目并不会删除磁盘上的相应文件。

查看项目中的依赖性

依赖性 (Dependencies) 可用于查看终端中 VI 所需的资源。每个终端均包括 **Dependencies**。

在 **Dependencies** 中无法直接添加条目。右键单击 **Dependencies** 并从快捷菜单中选择 **Refresh** 时，LabVIEW 可给终端中的 VI 添加相应依赖关系。例如，如果将包含子 VI 的 VI 添加到终端中，选择 **Refresh** 时 LabVIEW 会将对应的子 VI 添加到 **Dependencies** 中。但是如果在终端中添加依赖条目，该条目并不会在 **Dependencies** 中出现。例如，如果将 VI 和它的子 VI 都添加到终端中，选择 **Refresh** 时 LabVIEW 不会将子 VI 添加到 **Dependencies** 中。

依赖性包括 VI 静态调用的 VI、DLL、项目库等。



注

Dependencies 中并不包括 VI 动态调用的条目。必须将这些条目添加到终端中以便在项目中进行管理。

LabVIEW 可递归追踪子 VI，但无法递归追踪 DLL。例如，如果 a.vi 静态调用 b.dll 并且 b.dll 静态调用 c.dll，LabVIEW 仅将 b.dll 作为依赖条目。如需在项目中管理 c.dll，必须明确将 c.dll 添加到终端中。

如果依赖条目是项目库的一部分，LabVIEW 会将整个项目库添加到 **Dependencies** 中。

在 **Dependencies** 中无法创建新条目。也不能将 **Project Explorer** 中其它位置的条目拖放到 **Dependencies** 中。

如需删除 **Dependencies** 中的条目，可右键单击该条目并从快捷菜单中选择 **Remove**。如果使用的是这个方法删除条目，右键单击 **Dependencies** 并从快捷菜单中选择 **Refresh** 后该条目将会重新出现。

保存项目时，LabVIEW 并不会将依赖性作为项目的一部分进行保存。打开项目时，必须右键单击 **Dependencies** 并从快捷菜单中选择 **Refresh** 以查看相应依赖关系。

在项目中添加终端 (Windows)

Add Targets and Devices 对话框用于将终端或设备添加至项目。



注

必须安装可支持终端的模块或驱动程序以显示该对话框。

右键单击项目根并从快捷菜单中选择 **New»Targets and Devices** 以显示 **Add Targets and Devices** 对话框。如果该项目中的终端可支持其它终

端，也可以右键单击该终端并从快捷菜单中选择 **New»Targets and Devices**，从而在现有终端中添加其它终端。例如，如果在计算机中已安装 NI PCI 设备，可在 **My Computer** 终端中添加该设备。

为终端部署文件

在 **Project Explorer** 窗口中，打开和运行终端中的 VI，将文件部署至该终端中。运行该 VI 时，LabVIEW 可把相应的源文件、编译结果和硬件配置信息部署到终端上。

在多个终端中使用选板

LabVIEW 可为各个终端加载独立的选板。如果在项目的多个终端中均打开 VI，内存中将存在多个选板。在终端间进行切换时，LabVIEW 会保持各个锁定的选板与活动终端中对应选板的映射关系。如果锁定的选板没有对应的选板，对于该活动终端 LabVIEW 将显示最上层的选板。

运行程序实例

LabVIEW 会为项目中的各个终端创建一个程序实例或 LabVIEW 实例。在 **Project Explorer** 窗口打开 VI 时，该 VI 将在用于该终端的程序实例中打开。LabVIEW 也包括一个主程序实例，用于运行不属于项目或未从项目中打开的 VI。

通过在前面板和程序框图窗口的左下角所显示的程序实例名称，可确定 VI 所属的程序实例。如果已打开多个项目，程序实例名称将包括项目名称和终端名称，如 **Project 1.lvproj/My Computer**。否则程序实例名称将仅包括终端名称。

在 **Project Explorer** 窗口删除已打开的 VI 时，该 VI 将保存在原程序实例中。如果关闭并重新打开该 VI，LabVIEW 将在当前程序实例中打开 VI。

在多个程序实例中编辑 VI

可以同时多个程序实例中打开磁盘上的同一个 VI。例如，可以从两个项目或一个项目的两个不同终端中同时打开同一个 VI。

如果仅在一个程序实例中修改某个 VI，然后通过其它程序实例打开该 VI，所打开的 VI 将包括最新改动。但是如果所编辑 VI 已被多个程序实例打开，LabVIEW 不会在其它程序实例中自动更新改动。只有在所有程序实例中的 VI 一致时，才可以在其它程序实例中编辑、运行或保存该 VI。

通过以下方法可在所有程序实例中保持 VI 的一致性：

- 在 VI 工具栏中单击 **Synchronize with Other Application Instances** 以便在所有程序实例中更新 VI 改动。
- 在包含改动的程序实例中保存 VI 改动。保存 VI 时，LabVIEW 会自动更新所有其它程序实例的 VI 改动。
- 在包含改动的程序实例中撤销 VI 改动。



注

保存或同步 VI 后将无法撤销 VI 改动。

如果在其它程序实例中该 VI 正在运行或已进行执行保留，则无法编辑该 VI。如果 VI 是项目库的一部分，运行该 VI 时 LabVIEW 也会在所有程序实例中临时锁定该项目库。VI 在运行时无法编辑项目库，VI 停止运行后，LabVIEW 会为该项目库解锁。

在项目中使用的 VI 网络服务器

通过网络服务器可以查看项目中的 VI 或前面板。选择 **Operate»Connect to Remote Panel** 时，请尽量在 **VI Name** 中包括项目名、项目库和终端。

例如，如果 MyVI.vi 位于项目 MyProject.lvproj 的 My Computer 终端中，那么便可输入 MyProject.lvproj/My Computer/MyVI.vi 作为 **VI Name**。如果该 VI 位于名为 MyLibrary 的项目库中，那么也可以在 **VI Name** 中包括该项目库，如 MyProject.lvproj/My Computer/MyLibrary:MyVI.vi。如果项目或项目库中不包含该 VI，那么填写 **VI Name** 时无需任何附加信息。

LabVIEW 项目库

LabVIEW 项目库中可包括 VI、自定义类型 (type definitions)、共享变量 (shared variables)、选板菜单文件 (palette menu files)、其它文件和其它项目库等。创建并保存新项目库时，LabVIEW 会创建一个项目库 (project library) 文件 (.lvlib)，其中包括项目库属性以及项目库所包括的文件引用等。

使用项目库可以得到以下这些好处：以简单的条目层次结构管理文件，避免可能的 VI 名称重复、限制某些文件的公共访问权限、限制部分文件的编辑权限以及设置一组 VI 的默认选板菜单等。

通过 **Project Explorer** 窗口或独立运行的项目库窗口可以查看项目库的结构。如果尚未打开 **Project Explorer** 窗口，请查找一个项目库文件并在项目库窗口中打开。

在一个 LabVIEW 项目中可以创建一个项目库。右键单击 **My Computer** 并从快捷菜单中选择 **New»Library**。LabVIEW 所创建的项目库文件位于 **My Computer** 中。

也可以从项目中的文件夹创建项目库。在 **Project Explorer** 窗口中，右键单击文件夹并从快捷菜单中选择 **Convert to Library**。

项目库可用于组织一个合理的虚拟条目层次结构。项目库文件与 LLB 不同，LLB 包含 VI 的实际文件，而项目库文件并不包括实际文件。项目库所包括的文件仍以单独文件的形式保存在磁盘中。项目库所包括的文件组织结构可能与磁盘中的文件结构不同。

使用项目库可限定 VI 和其它 LabVIEW 文件的名称。LabVIEW 通过文件名识别 VI。正如交联 (cross-linking) 问题一样，LabVIEW 可能由于 VI 重名而错误地加载或引用内存中的 VI。当一个 VI 为项目库的一部分，LabVIEW 通过项目库名称来确保 VI 名称合法，以避免交联问题。合法文件名应包括文件名和包含该文件的项目库文件名。

例如，如果所创建的名为 `caller.vi` 的 VI 已经包含有存在于 `library1.lvlib` 中的名为 `init.vi` 的子 VI，它仍然可以调用 `library2.lvlib` 中的名为 `init.vi` 的子 VI，而不会发生交联问题。LabVIEW 所记录的 `caller.vi` 的合法文件名分别为 `library1.lvlib:init.vi` 和 `library2.lvlib:init.vi`。



注意

必须右键单击该项目库并从快捷菜单中选择 **Rename** 重命名项目库。如果没有通过 LabVIEW 重命名项目库，可能会断开与该项目库的连接。

右键单击项目库并从快捷菜单中选择 **Properties**，可以限制某些类型文件的公共访问权限或项目库的编辑权限，以及设置一个选板菜单文件 (.mnu) 用作项目库中所有 VI 的默认选板菜单。

设置默认选板菜单后，可以右键单击调用该项目库中任意 VI 的子 VI 实例，并从快捷菜单中查看该项目库的默认选板。

利用共享变量实现实时数据共享

在 LabVIEW 7.x 和前期版本中，对于无法使用连线连接的应用程序，通常通过局部和全局变量传递信息。根据该应用程序的范围，可以选择所使用变量的类型。LabVIEW 8.0 中所提出的共享变量可用于在项目中或跨网络的不同 VI 之间读写实时数据 (Live Data)。

共享变量包括现有 LabVIEW 数据传输技术 (如 DataSocket) 的各种功能，并可通过 **Project Explorer** 窗口进行管理。在前面板或程序框图均可读写共享变量数据。

共享变量属于配置好的软件项，用于在 VI 间传递数据。在不同 VI 或应用程序中无法使用连线进行连接的位置之间均可通过共享变量来共享数据。一个共享变量可表示单个值或一个 I/O (输入输出) 点。在使用共享变量的 VI 中，无需编辑程序框图即可修改该变量的范围 (scope)、**变量类型 (Variable Type)** 或其它共享变量属性。



注

必须对防火墙或网络地址转换 (Network Address Translating)(NAT) 路由器进行相应配置，才可通过防火墙或路由器传递共享变量。

创建共享变量

创建共享变量之前必须先打开一个项目。要在项目中添加共享变量，右键单击 **Project Explorer** 窗口中的终端 (target)、项目库或项目库 (project library) 中的文件夹，并从快捷菜单中选择 **New»Variable**。

共享变量必须包括在项目库中，如果从项目库以外的终端或文件夹创建共享变量，LabVIEW 会创建相应的新项目库并在项目库中包括该共享变量。

利用共享变量引用句柄 (Shared Variable Refnum) 和变量属性可通过编程配置项目中的共享变量。从 Variable Reference 属性创建一个显示控件即可创建一个共享变量引用句柄。但是不能使用该引用句柄修改单进程共享变量。

在前面板中读写共享变量值

通过前面板数据绑定 (front panel data binding) 可读写前面板对象中的实时数据。



注 前面板对象仅可与通过网络发布的共享变量绑定。

将 **Project Explorer** 窗口中的共享变量拖放至 VI 的前面板，可以创建该共享变量的控制范围 (control bound)。也可以右键单击一个控件，并从快捷菜单中选择 **Properties**，设置 **Data Binding** 选项卡上的相应选项，将控件与一个共享变量或网络中的 NI Publish-Subscribe Protocol (NI-PSP) 数据进行绑定。



注 在 LabVIEW 8.0 中，可以右键单击某个控件，并从快捷菜单中选择 **Data Operations»DataSocket Connection**。然后从 **Properties** 控制对话框的 **Data Binding Selection** 下拉菜单中选择 **DataSocket**，而不是通过 dstp、opc、ftp、http 和 file 协议连接前面板对象。

启用控件的数据绑定时，通过修改控件的值可改变与该控件绑定的共享变量的值。

在程序框图中读写共享变量值

共享变量 (Shared Variable) 节点是一个程序框图对象，用于指定 **Project Explorer** 窗口中相应的共享变量。共享变量节点可用于读写共享变量的值，并读取用于该共享变量数据的时间标识 (timestamp)。将 **Project Explorer** 窗口中的共享变量拖放至相同项目中 VI 的程序框图，可以创建一个共享变量节点。

也可以从**函数**选板上选择一个共享变量节点放置在程序框图上。要将程序框图中的共享变量节点与处于活动状态的项目中的共享变量进行绑定，可双击该共享变量节点显示 **Select Variable** 对话框，也可以右键单击该共享变量节点并从快捷菜单中选择 **Select variable**。

将共享变量限制为仅允许单个写操作

默认情况下，多个应用程序都可以对共享变量进行写操作。不过可以将通过网络发布的共享变量设置为每次仅从一个终端接受变量值的改动操作。在 **Shared Variable Properties** 对话框的 **Variable** 选项卡中勾选 **Single Writer** 复选框，可以确保共享变量每次仅允许一个写操作。共享变量引擎

(Shared Variable Engine) 仅允许对一台计算机的单个 VI 进行写操作。连接到共享变量的第一个程序可以进行写操作，而后来的程序无法进行写操作。第一个程序断开连接时，队列中的下一个程序将获得共享变量的写权限。对于无法对共享变量进行写操作的程序，LabVIEW 会发出相应提示。

启用共享变量缓冲

如果使用共享变量、psp 或 dstp URL 等方法通过编程共享数据，在默认情况下，LabVIEW 仅为所有读取该数据的程序提供数据的最新值。当客户端向服务器写入数据的速度比另一个客户端读取数据的速度快，新数据将在另一客户端读取数据之前覆盖较旧的未处理的数据。在读取数据的客户端收到后续数据值之前，如果没有收到该数据值，表示该数据将丢失。在服务器或客户端均可能发生未处理数据的丢失。如果正在读取数据并希望仅接收最近写入服务器的数据值，丢失未处理数据的问题并不会产生影响。然而，如果希望接收写入服务器的每一个共享变量值或需要无损耗数据传输，必须在客户端缓冲数据。在 **Shared Variable Properties** 对话框的 **Variable** 选项卡中，勾选 **Use Buffering** 复选框可以启用共享变量的数据缓冲。

管理共享变量

选择 **Tools»Shared Variable»Variable Manager** 可显示 **Variable Manager** 对话框。该对话框可用于在项目环境之外编辑、创建和监控共享变量。也可以使用对话框管理共享变量引擎 (Shared Variable Engine) 的生存周期。

创建源代码发布

前期版本中的 **Save With Options** 对话框中所包括的 VI 发布选项，如删除程序框图和设置密码等，现可在创建源代码发布 (Source Distribution) 时使用。

通过 **Project Explorer** 窗口的 **Build Specifications** 可创建源代码发布的程序生成规范。右键单击 **Build Specifications** 并从快捷菜单中选择 **New»Source Distribution**，可显示 **Source Distribution Properties** 对话框。

Application Builder 前期版本更改

Application Builder(应用程序生成器) 已集成到 **Project Explorer** 窗口中。如果使用 LabVIEW 基础版或完整版开发系统，请登录 National Instruments 网站 ni.com 并输入信息代码 rd1v21，单独购买 Application Builder。

通过 **Project Explorer** 窗口中的 **Build Specifications** 节点可创建独立应用程序 (EXE)、动态链接库 (DLL) 和压缩文件等的程序生成规范。**(Windows)** 也可以使用 **Build Specifications** 来创建安装程序的生成规范。

程序生成规范等同于应用程序生成器前期版本中的 .bld 文件，但现已成为 LabVIEW 项目的一部分而不是单独文件。



注

使用应用程序生成器工具之前必须创建一个项目。在打开的 VI 中使用 **Build Executable Wizard** 可以创建用于应用程序的项目。选择 **Tools»Build Executable** 运行 **Build Executable Wizard**，该向导用于指导整个创建过程。

在新项目中可以将 .bld 文件转换为程序生成规范。选择 **Tools»Convert Build Script** 查找并选择需转换的 .bld 文件。LabVIEW 可使用该文件创建一个包含源文件和构架规范的项目。

LabVIEW 8.0 包含以下应用程序生成器改进和修订：

- 各种类型的程序生成规范均包括一个多页面对话框，可用于配置和编辑生成规范。右键单击 **build specification** 并从快捷菜单中选择 **Properties**，即可打开该对话框。
- **(Windows)** 安装程序生成规范和其它生成规范有所不同。在一个安装程序构建中可包括多个应用程序或动态链接库构建。右键单击安装程序生成规范，并从快捷菜单中选择 **Properties**，转到 **Source Files** 选项卡并选择需添加至安装程序的应用程序、动态链接库和源代码发布构建等。
- 如果在程序生成规范中启用调试功能，在创建应用程序或动态链接库之前即可调试该程序和动态链接库。右键单击程序生成规范，并从快捷菜单中选择 **Properties**，转到 **Advanced** 选项卡并勾选 **Enable debugging** 复选框。



注

调试应用程序或动态链接库时，无法调试 Open VI Reference 函数创建的可重入 (reentrant) 前面板。如果可重入前面板是 LabVIEW 创建的动态链接库的接入点，也无法对其进行调试。

- **(Windows)** 在应用程序和动态链接库程序生成规范中指定一个版本号。右键单击应用程序生成规范，并从快捷菜单中选择 **Properties**，转到 **Application Information** 选项卡并设置程序版本号。右键单击动态链接库生成规范，并从快捷菜单中选择 **Properties**，转到 **Shared Library Information** 选项卡并设置动态链接库版本号。
- **(Windows)** 在安装程序构建中可以包括 NI 产品的安装程序或驱动程序。右键单击安装程序生成规范，并从快捷菜单中选择 **Properties**，转到 **Additional Installers** 选项卡并将所需驱动程序、LabVIEW Run-Time Engine 和其它 NI 产品安装程序添加至安装程序构建中。
- 在程序生成规范中可以配置应用程序或动态链接库的运行语言。右键单击应用程序或动态链接库生成规范，并从快捷菜单中选择 **Properties**，转到 **Run-Time Languages** 选项卡并设置相应语言选项。
- LabVIEW 运行引擎 (Run-Time Engine) 可支持多种语言，因此无需在应用程序、动态链接库或安装程序中包括多版本运行引擎。

- **(Windows)** 通过配置安装程序生成规范，可在安装程序中包括注册表项并且为安装程序中的文件创建多个快捷方式。右键单击安装程序生成规范，并从快捷菜单中选择 **Properties**，转到 **Registry** 选项卡并创建和配置自定义注册表项。通过 **Shortcuts** 选项卡可创建和配置快捷方式。
- 如果创建安装程序的应用程序或动态链接库中有 Storage VI，必须在该程序中包含 NI USI 安装程序。右键单击安装程序生成规范，并从快捷菜单中选择 **Properties**，转到 **Additional Installers** 选项卡并从 **National Instruments Installers to Include** 列表中选择 **NI USI**，即可包括 NI USI 安装程序。

新选板布局

LabVIEW 中改进了以下**控件 (Controls)** 和**函数 (Functions)** 选板：

- 选板条目现已根据类别进行排列。**标准 (Standard)** 和**标准 (图表或文本) (Standard (Icons or Text))** 格式已分别替换为**类别 (标准) (Category (Standard))** 和**类别 (图标和文本) (Category (Icons and Text))** 格式。下表列出 LabVIEW 7.1 中**控件**选板的顶层选板以及它们在 LabVIEW 8.0 中对应类别的位置。

控件选板

表 2 控件选板布局

7.1 选板	8.0 类别
快速 (Express)	快速 (Express)
数值 (Numeric)	新式 (Modern)
布尔 (Boolean)	新式 (Modern)
字符串 & 路径 (String & Path)	新式 (Modern)
数组 & 簇 (Array & Cluster)	新式 (Modern)
列表 & 表格 (List & Table)	新式 (Modern)
图形 (Graph)	新式 (Modern)
单选 & 枚举 (Ring & Enum)	新式 (Modern)
容器 (Containers)	新式 (Modern)
I/O	新式 (Modern)
引用句柄 (Refnum)	新式 (Modern)
修饰 (Decorations)	新式 (Modern)
对话框控件 (Dialog Control)	系统 (System)
经典控件 (Classic Control)	经典 (Classic)

函数选板

表 3 函数选板布局

7.1 选板	8.0 类别
快速 (Express)	快速 (Express)
结构 (Structures)	编程 (Programming)
数值 (Numeric)	编程 (Programming)
布尔 (Boolean)	编程 (Programming)
字符串 (String)	编程 (Programming)
数组 (Array)	编程 (Programming)
簇 (Cluster)	编程 (Programming)
比较 (Comparison)	编程 (Programming)
时间 & 对话框 (Time & Dialog)	编程 (Programming)
文件 I/O (File I/O)	编程 (Programming)
NI 测量 (NI Measurements)	测量 I/O (Measurement I/O)
波形 (Waveform)	编程 (Programming)
分析 (Analyze)	数学和信号处理 (Mathematics and Signal Processing)
仪器 I/O (Instrument I/O)	仪器 I/O(Instrument I/O)
应用程序控制 (Application Control)	编程 (Programming)
图形 & 声音 (Graphics & Sound)	编程 (Programming)
通信 (Communication)	跨平台连接和数据通信 (Connectivity and Data Communication)
报表生成 (Report Generation)	编程 (Programming)
高级 (Advanced)	跨平台连接和数据通信 (Connectivity and Data Communication)
修饰 (Decorations)	编程 (Programming)

- 在**类别 (标准) (Category (Standard))** 和**类别 (图标和文本) (Category (Icons and Text))** 格式中，右键单击某个类别并从快捷菜单中选择 **Move this Category Up** 或 **Move this Category Down**

可以改变选板中类别的排列顺序。也可以单击类别左侧的双直线并将箭头拖放到显示该类型的新位置。

- 默认情况下，除了从 LabVIEW 前期版本进行升级并且该版本中已使用 **高级 (Advanced)** 选板视图外，LabVIEW 安装后都将显示精简选板视图。精简选板视图将在最高层选板中隐藏部分高级类别。单击选板工具栏 (palette toolbar) 中的 **View** 按钮并从快捷菜单中选择 **Always Visible Categories»Show All Categories**，显示 **控件** 或 **函数** 选板中的所有类别。
- 当前选板或所有选板均可以改变格式。单击 **控件** 或 **函数** 选板工具栏中的 **View** 按钮，并从 **View This Palette As** 快捷菜单中选择所需格式可改变当前选板的格式。选择 **Tools»Options**，在 **Controls/Functions Palettes** 选项卡中可给所有选板选择一种格式。
- **树形格式 (Tree format)** — 在树形控件中以文本的形式显示选板项目。单击 **控件** 或 **函数** 选板工具栏中的 **View** 按钮，并从快捷菜单中选择 **View This Palette As»Tree** 可将当前选板设置为树形格式。
- 在 **文本** 和 **树形格式** 中，单击 **控件** 或 **函数** 选板工具栏中的 **View** 按钮，并从快捷菜单中选择 **Sort Alphabetically** 可将位于同级的项目按字母顺序进行重排。
- 使用 **收藏 (Favorites)** 类别可将 **函数** 选板中的常用项目集合在一起。通过 **Category (Standard)**、**Category (Icons and Text)**、**Icons** 和 **Icons and Text** 格式可以将条目添加至 **Favorites** 类别。在固定的 **函数** 选板上，右键单击对象并从快捷菜单中选择 **Add Item to Favorites**，可以将该对象添加到 **Favorites** 类别中。在 **Category (Standard)** 和 **Category (Icons and Text)** 格式中，也可以展开一个选板以显示其子选板，右键单击该子选板的标题并从快捷菜单中选择 **Add Subpalette to Favorites**。

已移动和重命名的选板

在 LabVIEW 8.0 中已移动或重命名以下选板。

- 在 **VISA Advanced** 选板中，**Interface Specific** 和 **High Level Register Access** 选板已分别更名为 **Bus/Interface Specific** 和 **Register Access** 选板。
- **Advanced Formula Parsing** 选板中的 VI 已转移至 **Formula Parsing** 和 **1D & 2D Evaluation** 选板中。
- **Cluster** 和 **Variant** 选板中的 VI 已转移至 **Cluster & Variant** 选板。
- **Array Operations**、**Curve Fitting**、**Formulas**、**Logarithmic** 和 **Special and Numeric Functions** 选板已分别更名为 **Signal Operation**、**Fitting**、**Scripts & Formulas**、**Exponential Functions** 和 **Elementary & Special Functions** 选板。部分曲线拟合 (Curve Fitting) VI 现已位于 **Interp & Extrap** 选板中。
- **Express Numeric Constants** 选板已更名为 **Express Math & Scientific Constants** 选板。

- **Additional Numeric Constants** 选板已更名为 **Math & Scientific Constants** 选板。
- **Express Logarithmic** 选板已更名为 **Express Exponential Functions** 选板。
- **Array Operations PtByPt** 选板中的 VI 现已位于 **Geometry PtByPt**、**Polynomial PtByPt**、**Probability & Statistics PtByPt** 和 **Signal Operation PtByPt** 选板中。
- **Curve Fitting PtByPt** 选板中的 VI 现已位于 **Fitting PtByPt** 和 **Interpolation PtByPt** 选板中。
- **Advanced** 选板已更名为 **Libraries & Executables** 选板。
- **Time & Dialog** 选板中的 VI 现已位于 **Timing** 和 **Dialog & User Interface** 选板中。
- **Frequency Domain** 选板中的 VI 现已位于 **Spectral Analysis** 和 **Transforms** 选板中。
- **Frequency Domain PtByPt** 选板中的 VI 现已位于 **Spectral Analysis PtByPt** 和 **Transforms PtByPt** 选板中。
- **Time Domain** 选板中的 VI 现已位于 **Signal Operation** 和 **Integ & Diff** 选板中。
- **Time Domain PtByPt** 选板中的 VI 现已位于 **Signal Operation PtByPt** 和 **Integral & Differential PtByPt** 选板中。
- **Timed Loop** 选板现已更名为 **Timed Structures** 选板。
- **函数** 选板中不再包括 **All VIs and Functions** 选板。

选板编辑改进

LabVIEW 8.0 在编辑选板上作了以下改进：

- 在 LabVIEW 7.x 和前期版本中，可以创建或编辑**控件**或**函数**选板的自定义视图。LabVIEW 8.0 不再支持自定义选板视图。现在无需使用自定义选板视图即可编辑选板。如果同时操作多个终端，LabVIEW 会在各个终端上加载独立的选板。编辑特定终端的选板时，在需要进行修改终端的 VI 中打开 **Edit Controls and Functions Palette Set** 对话框。选择 **Tools»Advanced»Edit Palette Set**，显示 **Edit Controls and Functions Palette Set** 对话框。
- LabVIEW 在默认数据目录中可将选板的改动保存至 8.0\Palettes 文件夹。
- 如需隐藏与目录同步的项目，可使用右键单击该项目并从快捷菜单中选择 **Hide Synchronized Item**。
- 如果项目已与目录进行同步，通过右键单击该项目并从快捷菜单中选择 **Display Synchronization Path** 可显示源目录的路径。
- 如果位于选板末端的行或列仅包括已被隐藏的同步项目和/或 LabVIEW 无法找到的项目，只有在编辑选板情况下才可显示该行或列。
- 如果子选板为空或仅包含 LabVIEW 无法查找的项目，只有在编辑选板情况下才可显示该子选板。

菜单重组

LabVIEW 添加了部分新菜单，并且删除或重命名了一些菜单选项。以下 LabVIEW 菜单选项已修改：

- **文件 (File)** — 添加了基本项目操作的菜单选项，如创建 (creating)、打开 (opening)、保存 (saving) 和关闭 (closing) 选项。
- **编辑 (Edit)** — 已将部分 **Operate** 菜单选项转移至 **Edit** 菜单中，并增加了新的 **Select All** 选项。
- **查看 (View)** — 已添加一个新菜单，所包括选项可用于显示窗口和选板、浏览类和显示 **Project Explorer** 窗口。**窗口 (Window)** 部分菜单选项现已更改为 **View** 的部分菜单。由于已删除 **Browse** 菜单选项，因此 **View** 菜单将包括 **Browse** 的选项。选择 **View»Browse Relationships** 可找到 **Browse** 菜单选项。
- **项目 (Project)** — 已添加一个新菜单，所包括选项可用于在项目中添加 VI 和文件、创建属性和查看项目信息。
- **操作 (Operate)** — 已将用于调试 VI 的选项添加或转移至 **Operate** 菜单中。由于已删除 **Browse** 菜单选项，因此 **Operate** 菜单将包括 **Browse** 的部分选项。
- **工具 (Tools)** — 已将部分选项添加至 **Tools** 菜单中。已重新安排部分现有 **Tools** 菜单选项至不同子菜单中。
- **窗口 (Window)** — 已删除用于查看选板的选项，如 **Navigation** 窗口和 **Error list** 窗口。上述选项均属于 **View** 菜单。
- **帮助 (Help)** — 已添加用于查找仪器驱动程序和激活 LabVIEW 的选项。



注

部分菜单选项只在某些情况中有效，比如在特定操作系统中使用 LabVIEW 开发系统特殊版本、已选择 **Project Explorer** 窗口中的选项或已选定某个 VI 时。

关于性能和内存问题的详细信息见 *LabVIEW Help*。

使用源代码控制

如果已安装第三方源代码控制软件，如 Perforce 或 Microsoft Visual SourceSafe，可在 LabVIEW 项目中或在不属于项目的 VI 中使用源代码控制 (Source Control)。在 VI、项目中的文件以及项目文件（如项目库）中可以实现源代码控制操作。LabVIEW 所提供的基础结构可用于连接外部第三方源代码控制软件。必须安装第三方源代码控制软件才可以在 LabVIEW 中使用源代码控制。**(Windows)** LabVIEW 可集成任何支持 Microsoft 源代码控制接口 (Source Code Control Interface) 的源代码控制软件。**(Mac OS 和 Linux)** 可以使用 Perforce 命令行界面。

LabVIEW 8.0 包括以下源代码控制功能改动。



注

仅可在 LabVIEW 专业版开发系统中使用源代码控制。

- 在LabVIEW 8.0中已无法使用LabVIEW 7.x和前期版本的内置源代码控制软件。关于从 LabVIEW 前期版本转移源代码控制文件的详细信息见从 LabVIEW 内置源代码控制软件中进行转移一节。
- 选择并安装第三方源代码控制软件后，必须配置 LabVIEW 以便能够正常运行该软件。选择 **Tools»Source Control»Configure Source Control** 可显示 **Options** 对话框的 **Source Control** 选项卡。
- 在 LabVIEW 中使用源代码控制可实现以下功能：提取 (check out) 需进行编辑的文件或提交 (check in) 已编辑的文件、从源代码控制中获取某个文件的最新版本以及将某个文件与源代码控制中的最新版本进行比较等。单击 **Tools»Source Control** 并选择所需的源代码控制操作。
- 也可查看某个文件的源代码控制历史。选择 **Tools»Source Control»Show History** 显示特定软件的对话框，其中包含源代码控制中的文件修改的历史信息。
- Source Control VI 可用于通过编程实现源代码控制操作。
- 在 LabVIEW 中已无法创建 Perforce 配置文件。

从 LabVIEW 内置源代码控制软件中进行转移

在 LabVIEW 8.0 中已无法使用 LabVIEW 7.x 和前期版本的内置源代码控制软件。必须选择第三方源代码控制软件才可以在 LabVIEW 中使用源代码控制。如果在前期版本中已使用内置软件，必须将文件转移至其它源代码控制软件中以便继续在 LabVIEW 中使用源代码控制。关于 LabVIEW 可支持的第三方源代码控制软件列表，请登录 National Instruments 网站 ni.com 并输入信息代码 `rdbp01` 进行查询。

将文件转移至新的源代码控制软件时，将会丢失内置软件所保存的修改历史。文件的前期版本不能转移至新软件中。

请按以下步骤将文件从内置源代码控制软件中转移至第三方源代码控制软件。

1. 在 LabVIEW 前期版本中，确定 LabVIEW 内置源代码控制软件所包含的文件已经全部被提交 (check in)。
2. 在需将文件添加至新的源代码控制软件的计算机中，使用内置软件获取所有文件的最新版本。
3. 通过内置软件从源代码控制中提取 (check out) 文件。
4. 在第三方源代码控制软件中，配置新源代码控制项目所需的设置。
5. 配置 LabVIEW 以兼容第三方源代码控制软件。
6. 创建一个 LabVIEW 项目。将内置软件所包含的文件添加至该项目中。LabVIEW 跳出提示信息时，将文件添加至源代码控制中，也可以直接在第三方软件中添加文件。

VI 和函数的改进

LabVIEW 8.0 包含以下现有 VI 和函数的改动。

Analyze VI 改进

在 LabVIEW 7.x 和前期版本中，**Analyze** 选板上的 VI 现已分为以下三个单独选板 **Mathematics**、**Signal Processing** 和 **Programming**。某些 VI 的图标已更改，各个 VI 图标顶部的黄色标志用于表示包括该 VI 所属的子选板。

已改进或修改的 VI

在 LabVIEW 8.0 中，已分别改进或修改以下 **Mathematics**、**Signal Processing** 和 **Programming** VI。

数学选板 (Mathematics Palette)

Fitting VI 包含以下改进和修订：

- Exponential Fit、Exponential Fit Coefficients、Linear Fit 和 Linear Fit Coefficients VI 的 **method** 输入端可确定所使用的拟合方法。现可使用 Least Square、Least Absolute Residual 或 Bisquare fitting 方法。上述 VI 也包括一个 **tolerance** 输入端，如果所选择方法要求迭代 (iteration)，该输入可用于决定停止对输出参数进行迭代调整的时机。上述 VI 的 **Weight** 输入端可确定所观测数据值的权值大小。
- Exponential Fit 和 Linear Fit VI 的 **residue** 输出端可返回该拟合模型 (fitted model) 的加权平均误差。
- Exponential Fit 和 Exponential Fit Coefficients VI 的 **refine?** 输入端可确定是否进一步提高该拟合模型的 **振幅 (amplitude)** 和 **衰减 (damping)** 的精度。
- Exponential Fit VI 不再包括 LabVIEW 7.x 和前期版本提供的 **Standard Deviation** 输入端或 **mse** 输出端。
- Linear Fit VI 不再包括 LabVIEW 7.x 和前期版本提供的 **mse** 输出端。
- General LS Linear Fit VI 的 **Weight** 输入端可确定所观测数据的权值大小。该 VI 提供的新算法 **SVD for Rank Deficient H** 可提高计算精度。建议仅当输入矩阵是不满秩矩阵并且所有其它算法均失败时，才尝试使用该算法。该 VI 不再包括 LabVIEW 7.x 和前期版本提供的 **Standard Deviation** 输入端。
- General Polynomial Fit VI 的 **Coefficient Constraint** 输入端可指定对某特定阶数下的各项式系数的约束。该 VI 的 **Polynomial Fit Coefficients** 输出端现已更名为 **Polynomial Coefficients**。

Integ & Diff VI 包含以下改进和修订：

- Numeric Integration VI 可实现二维和三维数值积分 (numeric integration)。

Linear Algebra VI 包括以下改进和修订：

- 在 LabVIEW 7.x 中，Eigenvalues and Vectors VI 的 **eigenvector** 输出端已进行归一化，使得其中的最大分量为 1。在 LabVIEW 8.0 中，对 **eigenvector** 的归一化使得它的 Euclidean 范数等于 1。
- QR Decomposition VI 将取代 QR Factorization VI。QR Decomposition VI 包括两个新输入端。**pivot?** 可指定 VI 是否对输入矩阵 **A** 进行列选主元的分解。**Q option** 可指定 VI 产生正交矩阵 (orthogonal matrix) **Q** 的方式。该 VI 可产生实际大小 (full-size)、经济型大小 (economy-size) 的正交矩阵或不产生正交矩阵。该 VI 的 **P** 输出端可返回置换矩阵 (permutation matrix)。
- QZ Decomposition VI 包括两个新输入端。**decomposition type** 可指定是否执行广义 Hessenberg 或广义 Schur 分解，**order** 可确定对广义特征向量、**Alpha** 和 **Beta** 进行排序的方式。
- Schur Decomposition VI 的 **order** 输入端用于确定对 **Eigenvalues** 和相应的 **Schur Form** 和 **Schur Vectors** 进行排序的方式。

Polynomial VI 包含以下改进和修订：

- Polynomial Roots VI 可以查找复系数多项式 (complex polynomial) 的根，该 VI 也包括 **Advanced Refinement** 根选项用于得到更精确的根。
- Remove Zero Coefficients VI 的 **number of zero coefficients** 输出端可返回 LabVIEW 所删除的近似于零的高阶系数的个数。

Probability and Statistics VI 包含以下改进和修订：

- Mode VI 现为多态 VI，并可执行单峰 (unimodal) 和多峰 (multimodal) 分析。该 VI 的 **intervals** 输入端现支持零和负整数。如果 **intervals** 不是正整数，Mode VI 只计算输入序列的准确模式 (mode)。
- Standard Deviation and Variance VI 的 **X** 输入端现更改为推荐而不是必需项目。

Rational Polynomial VI 包含以下改进和修订：

- Pade Approximation VI 的 **c(0..n+m)** 输入端现已更名为 **Derivatives**，同时 **A(0..m)** 输出端和 **B(0..n)** 输出端已分别更名为 **Numerator** 和 **Denominator**。

信号处理选板 (Signal Processing Palette)

Advanced FIR Filtering VI 包含以下改进和修订：

- FIR Windowed Coefficients VI 的 **window parameter** 输入端可为 Kaiser、Gaussian 和 Dolph-Chebyshev 窗口指定专用的参数值。该 VI 的 **window** 输入端的数据类型已由枚举类型改为 32 位无符号整数。

Advanced IIR Filtering VI 包含以下改进和修订:

- Cascade → Direct Coefficients VI 现已更名为 Cascade To Direct Coefficients VI。
- IIR Cascade Filter、IIR Cascade Filter with I.C.、IIR Filter 和 IIR Filter with I.C. VI 现为多态 VI 并支持复数输入。

Filters VI 包含以下改进和修订:

- FIR Windowed Filter VI 的 **window parameter** 输入端可为 Kaiser、Gaussian 和 Dolph-Chebyshev 窗口指定专用的参数值。该 VI 的 **window** 输入端的数据类型已由枚举类型改为 32 位无符号整数。
- 在 **Filters** 选板中的所有 VI (不包括 Median Filter VI) 现均为多态 VI 并可支持复数输入。

Signal Generation VI 包含以下改进和修订:

- Binary MLS VI 现已包括一个 **error code** 输出端。
- Ramp Pattern VI 的 **exclude end?** 输入端可确定是否在所产生数组中包括最终值。
- Signal Generator by Duration VI 的 **signal type** 输入端现更改为建议连接而不是必需连接。

Signal Generation PtByPt VI 包含以下改进和修订:

- Sawtooth Wave PtByPt VI 已进行优化并使用一个新的方程, 如下所示:

$$\text{sawtooth wave} = \text{amplitude} \times \text{sawtooth}(q)$$

其中

$$\text{sawtooth}(q) = \begin{cases} \frac{q}{180} & \text{if } q < 180 \\ \frac{q}{180} - 2 & \text{else} \end{cases}$$

并且

$$q = (360 \cdot f \cdot \text{time}) + \text{phase} \bmod (360), \text{ 相位单位是度。}$$

Signal Operation VI 包含以下改进和修订:

- Autocorrelation、Convolution 和 CrossCorrelation VI 现为多态 VI 并支持二维实数和复数输入。
- Autocorrelation 和 CrossCorrelation VI 的一维实例的 **normalization** 输入端可确定所使用的归一化方法, 分别用于计算自相关和互相关。
- Convolution VI 的二维实例的 **output size** 输入端可确定卷积 (convolution) 的长度。

- Normalize 多态 VI 现包括 Normalize Vector VI 和 Normalize Matrix VI。
- Quick Scale 多态 VI 现包括 Quick Scale 1D VI 和 Quick Scale 2D VI。Quick Scale VI 的 Quick Scale 2D 实例的 $Y_{ij}=X_{ij}/\text{Max}\{X\}$ 参数现更名为 $Y_{ij}=X_{ij}/\text{Max}|X|$ 。
- Resample (constant to constant) 和 Resample (constant to variable) VI 现可支持复数输入。
- Scale 多态 VI 现包括 Scale 1D VI 和 Scale 2D VI。
- Unit Vector VI 现为多态 VI 并包括两个新输入端。**norm type** 表示用于计算范数所使用的范数类型。如果 **norm type** 是 **User Defined**，另一输入端 **user defined norm** 可用于确定范数类型。
- Zero Padder VI 现为多态 VI 并可支持复数输入。该 VI 的 **only non-power of 2?** 输入端用于确定当输入数组大小已经是 2 的幂时，是否将输入数组变成原来两倍。

Spectral Analysis VI 包含以下改进和修订：

- Power Spectrum VI 现为多态 VI 并可支持复数输入。
- STFT Spectrogram VI 现已更名为 STFT Spectrograms VI，并包括两个新输入端。**time-freq sampling info** 输入端可确定在联合时频域中采集信号所使用的密度，并指定所得到的二维时频数组的大小。**window info** 可确定用于计算短时傅立叶变换 (Short-Time Fourier Transform) 的窗口的信息。现在可以使用 Blackman-Harris、Exact Blackman、Flat Top、4 Term B-Harris、7 Term B-Harris、Low Sidelobe、Blackman Nuttall 和 Triangle 窗口。该 VI 不再包括 LabVIEW 7.x 和前期版本提供的 **time increment**、**window length** 或 **window selector** 输入端。

Transforms VI 包含以下改进和修订：

- FFT 和 Inverse FFT VI 现可支持二维数组实数和复数输入。
- FFT 和 Inverse FFT VI 的 **shift?** 输入端可确定是否将 DC 分量旋转至至快速傅立叶变换 (Fast Fourier Transform) 的中部。
- FFT VI 的一维实例的 **FFT size** 输入端可确定 FFT 计算的长度。

Transforms PtByPt VI 包含以下改进和修订：

- FFT PtByPt 多态 VI 现包括 Real FFT PtByPt VI 和 Complex FFT PtByPt VI。
- Inverse FFT PtByPt 多态 VI 现包括 Inverse Real FFT PtByPt VI 和 Inverse Complex FFT PtByPt VI。

Waveform Conditioning VI 包含以下改进和修订：

- Scaled Window VI 支持复数输入。该 VI 的 **window parameter** 输入端可为 Kaiser、Gaussian 和 Dolph-Chebyshev 窗口指定专用的参数值。该 VI 的 **window** 输入端的数据类型已由枚举类型改为 32 位无符号整数。

Waveform Measurements VI 包含以下改进和修订：

- 以下 VI 的 **window** 输入端的数据类型已由枚举类型转变为 32 位无符号整数：Cross Spectrum (Mag-Phase)、Cross Spectrum (Real-Im)、FFT Power Spectral Density、FFT Power Spectrum、FFT Spectrum (Mag-Phase)、FFT Spectrum (Real-Im)、Frequency Response Function (Mag-Phase) 和 Frequency Response Function (Real-Im) VI。

Windows VI 包含以下改进和修订：

- Cosine Tapered Window VI 的 **ratio** 输入端可确定渐变部分 (tapered section) 的长度与整个信号长度的比率。该 VI 的 **Cosine Tapered{X}** 输出端现已更名为 **Windowed X**。
- Scaled Time Domain Window VI 的 **window parameter** 可用于为某些窗口类型指定专用的参数值。该 VI 也包括一个新的 **error** 输出端。该 VI 的 **Waveform** 输入端、**Windowed Waveform** 输出端和 **window constants** 输出端现已分别更名为 **X**、**Windowed X** 和 **window properties**。Scaled Time Domain Window VI 也支持 Blackman-Nuttall、Triangle、Kaiser、Dolph-Chebyshev 或 Gaussian 窗口。该 VI 的 **window** 输入端的数据类型已由枚举类型改为 32 位无符号整数。
- 在 **Windows** 选板中的所有 VI (不包括 Window Properties VI) 现均为多态 VI 并可支持复数输入。

编程选板 (Programming Palette)

Comparison 函数包含以下改进和修订：

- Equal? 和 Not Equal? 比较函数可确定两个 VI Server 引用 (VI Server reference) 是否指向相同的对象。

新 VI 和函数

LabVIEW 8.0 将在选板中添加以下新 Mathematics、Signal Processing 和 Programming VI。

数学选板 (Mathematics Palette)

Advanced Curve Fitting 子选板包含下列新 VI：

- Exponential Fit Intervals VI
- Gaussian Peak Fit Coefficients VI
- Gaussian Peak Fit Intervals VI
- Goodness of Fit VI
- Linear Fit Intervals VI
- Logarithm Fit Coefficients VI
- Logarithm Fit Intervals VI
- Power Fit Coefficients VI

- Power Fit Intervals VI
- Remove Outliers VI

Differential Equations 子选板包括新 ODE Solver VI。

Discrete Math 子选板包含以下新 VI:

- Gcd VI
- Lcm VI
- Permute VI

Exponential Functions 子选板包括新 y-th root of x 函数。

Fitting 子选板包含以下新 VI:

- Cubic Spline Fit VI
- Gaussian Peak Fit VI
- Logarithm Fit VI
- Nonlinear Curve Fit VI
- Power Fit VI

Geometry 子选板包含以下新 VI:

- 2D Cartesian Coordinate Shift VI
- 2D Cartesian Coordinate Rotation VI
- 3D Cartesian Coordinate Shift VI
- 3D Cartesian Coordinate Rotation (Euler) VI
- 3D Cartesian Coordinate Rotation (Direction) VI
- 3D Coordinate Conversion VI
- Direction Cosines to Euler Angles VI
- Euler Angles to Direction Cosines VI

Hyperbolic Functions 子选板包括以下新函数:

- Hyperbolic Cosecant 函数
- Hyperbolic Cotangent 函数
- Hyperbolic Secant 函数
- Inverse Hyperbolic Cosecant 函数
- Inverse Hyperbolic Cotangent 函数
- Inverse Hyperbolic Secant 函数

Hypothesis Testing 子选板包含以下新 VI:

- Correlation Test VI
- Sign Test VI
- T Test VI

- Wilcoxon Signed Rank Test VI
- Z Test VI

Integ & Diff 子选板包含新 Lobatto Quadrature VI。

Interp & Extrap 子选板包含以下新 VI:

- Create Interpolating Polynomial VI
- Create Mesh Grid (2D) VI
- Evaluate Interpolating Polynomial VI
- Hermite Interpolation 1D VI
- Interpolate 1D VI
- Interpolate 1D Fourier VI
- Interpolate 2D VI
- Search Ordered Table VI
- Spline Interpolation 1D VI

Numeric 子选板包括以下新函数:

- Machine Epsilon 常量
- Square 函数

Optimization 子选板包含以下新 VI:

- Constrained Nonlinear Optimization VI
- Quadratic Programming VI
- Unconstrained Optimization VI

Polynomial 子选板包含新 Polynomial Plot VI。

Probability 子选板包含以下新 VI:

- Continuous CDF VI
- Continuous Inverse CDF VI
- Continuous Moments VI
- Continuous PDF VI
- Continuous Random VI
- Discrete CDF VI
- Discrete Inverse CDF VI
- Discrete Moments VI
- Discrete PF VI
- Discrete Random VI

Probability and Statistics 子选板包含以下新 VI:

- Correlation Coefficient VI
- Correlation Coefficient (Kendall's Tau) VI
- Correlation Coefficient (Spearman) VI
- Covariance Matrix VI
- Measures of Mean VI
- Measures of Spread VI
- Percentiles VI

Trigonometric Functions 子选板包括以下新函数:

- Inverse Cosecant 函数
- Inverse Cotangent 函数
- Inverse Secant 函数

信号处理选板 (Signal Processing Palette)

Filters 子选板包含新 Zero Phase Filter VI。

Signal Operation 子选板包含新 Riffle VI。

Transforms 子选板包含以下新 VI:

- Chirp Z Transform VI
- DCT VI
- DST VI
- Inverse DCT VI
- Inverse DST VI

Waveform Conditioning 子选板包含新 Continuous Convolution (FIR) VI。

Waveform Measurements 子选板包含新 Extract Multiple Tone Information VI。

Windows 子选板包含以下新 VI:

- Blackman-Nuttall Window VI
- Chebyshev Window VI
- Gaussian Window VI
- Symmetric Window VI
- Window Properties VI

编程选板 (Programming Palette)

Comparison 子选板包括以下新函数:

- Empty Array? 函数

- Polar to Re/Im 函数
- Re/Im to Polar 函数

Digital Waveform VI 改进

Digital Waveform VI 可用于替换数字波形 (DWDT) 或数字表格 (DTbl) 的一个子集，将数字波形数据或数字数据转换为模拟波形或二进制数据格式，或者将模拟波形、二进制格式、电子表格字符串转换为数字波形数据或数字数据。数字模式发生器 (Digital Pattern Generator) 可以生成一个包括多种填充模式的数字模式。使用 Empty Digital Data 和 Empty Digital Waveform 常数可以返回空的数字波形数据或数字数据，这对于初始化移位寄存器或创建波形非常有用。

已改进或修改的 VI

在 LabVIEW 8.0 中，已改进或修改以下 Digital Waveform VI。

- Invert Digital VI 也包括两个新输入端。**signal index** 可确定开始转换数据的信号，**number of signals** 可确定需进行转换的信号数量。
- Search for Digital Pattern VI 的 **compare mode** 输入端可确定搜索时该如何处理 X 的值。
- Analog to Digital VI 是多态 VI 并可将其模拟波形转换为数字数据。
- Digital to Analog VI 是多态 VI 并可将其数字数据转换为模拟波形。

新 VI 和函数

LabVIEW 8.0 包含以下新 Digital Waveform VI 和函数。

- Binary to Digital VI
- Digital Pattern Generator VI
- Digital Ring constant
- Digital to Binary VI
- Empty Digital Data constant
- Empty Digital Waveform constant
- Replace Subset VI
- Spreadsheet String to Digital VI

禁用结构

条件禁用 (Conditional Disable) 结构可用于禁用部分代码段。只要给结构的每个子程序框图指定一个包含符号和值的表达式 (由用户和 LabVIEW 所定义)，LabVIEW 即可利用表达式决定所使用的子程序框图。



注

LabVIEW 并不会编译程序框图中不会执行的代码。例如，Conditional Disable 结构的任何非活动子程序框图中存在的代码，LabVIEW 均不会对其进行编译。此外，对于已连接常数的条件结构 (Case structure)，如果某条件分支不被执行，LabVIEW 也不会编译这部分代码。

使用 Diagram Disable 结构可以定义不希望进行编译的部分程序框图代码。

Express VI 改进

ExpressVI 和函数可用于创建常用测量任务。

已改进或修改的 VI

在 LabVIEW 8.0 中，已改进或修改以下 Express VI：

- Comparison Express VI 包括两个新输入端。**Items to Compare** 可用于选择对输入信号的数据点、时间标识、数据点间的时间间隔、数据点数量以及信号名称进行比较。**Constant Value to Compare Against** 确定和 **Operand 1** 输入端进行比较的常数值。该 VI 也已经被优化，在第一个输入信号为空的时候仍可以返回合理结果。
- Read LabVIEW Measurement File 和 Write LabVIEW Measurement File Express VI 已分别更名为 Read From Measurement File 和 Write To Measurement File。上述 Express VI 可同时支持二进制测量文件 (.tdm) 和文本测量文件 (.lvm)。在配置对话框中选择 **Binary (TDM)** 选项可设置文件格式。
- Read From Measurement File、Write To Measurement File、Prompt User for Input 和 Report Express VI 根据不同终端修改相应的行为。如果当前终端未连接主机，配置对话框将在非法选项后显示警告信息。如果将 VI 配置为提示用户输入数据，但执行该 VI 的终端并没有用户界面（例如未连接主机的 LabVIEW 实时模块），上述 VI 将返回错误信息。
当 Report VI 运行于非 Windows 平台上时，配置对话框仅支持**用于网页的 HTML (HTML for Web Page)** 作为目标位置。
- 如果 **Enable** 输入端未连接任何值，Relay Express VI 将使用配置对话框指定的 **Enable** 值。
- Select Signals Express VI 的 **Selector Input** 输入端可允许用户从 **Signals** 输入中选择一个信号子集。
- Write Data 和 Write To Measurement File Express VI 已优化从而保证了通道名称的唯一性。当文件名或通道名称重复时，LabVIEW 将在名称末端追加一个整数以确保该名称的唯一性。例如，如果所提供名称包括 sine、sine、square、square 和 sine，LabVIEW 会将上述名称分别更改为 sine、sine 1、square、square 1 和 sine 2。

新 VI 和函数

LabVIEW 8.0 包含以下新 Express VI：

- Acquire Sound Express VI
- Dual Channel Spectral Measurement Express VI
- Play Waveform Express VI
- NI DIAdem Report Express VI

NI DIAdem Report Express VI

通过 NI DIAdem Report Express VI 可直接从 LabVIEW 数据生成报表。可将该报表导出为 PDF 或 HTML 文件，打印报表或者自动显示该报表。在 LabVIEW 中可完成对该报表的配置。



注

使用 NI DIAdem Report Express VI 之前，必须安装 DIAdem 9.1 Service Pack 2 或更高版本。关于下载 DIAdem 试用版或购买 DIAdem 的详细信息，请登录 National Instruments 网站 ni.com 进行查询。

File I/O VI 和函数改进

File I/O 选板包括已更新的 VI 以及最新的和已更新的函数，使文件读写数据操作更容易，其中包括对路径输入 (path input) 和新建对话框 (new datalog) 函数等的改进。**File I/O** 选板还包括用于处理压缩文件的新 VI。

路径输入

接受路径或引用句柄作为输入的 File I/O 函数，都包括一个名字中含有 (use dialog) 短语的输入端，例如 **path (use dialog)** 输入端。如果未对该输入端连线，当函数执行时对话框将提示用户选择文件或目录。每个函数均包括 **cancelled** 输出端。当用户单击对话框中 **Cancel** 按钮时，**cancelled** 返回 TRUE，函数返回错误。如果将路径连接到 File I/O 函数，将以最小权限来完成操作，从而无需显式地打开或关闭该文件。

Datalog 函数

数据记录 (Datalog) 函数可用于读写数据记录文件 (datalog file)。

Datalog 选板包括以下新函数：

- Open/Create/Replace Datalog 函数
- Get Datalog Position 函数
- Get Number of Records 函数
- Read Datalog 函数
- Set Datalog Position 函数
- Set Number of Records 函数
- Write Datalog 函数

File I/O VI 和函数

File I/O VI 和函数可用于打开或关闭文件、读写文件、创建目录或文件、获取目录信息以及将字符串、数字、数组和簇写入文件等。**File I/O** 选板包括以下新函数：

- Open/Create/Replace File 函数
- Read from Binary File 函数
- Read from Text File 函数

- Write to Binary File 函数
- Write to Text File 函数

Zip VI

Zip VI 可用于新建压缩文件、在压缩文件中添加其它文件以及关闭压缩文件。

其它改进和修订

LabVIEW 8.0 包括以下其它 File I/O 和函数的改进及修订：

- Copy 函数包括 **overwrite** 和 **prompt** 输入端以及一个 **cancelled** 输出端。
- Delete 函数包括 **entire hierarchy** 输入端，如果输入为 TRUE，函数会删除位于所指定 **路径 (path)** 或路径下的所有文件。
- File Dialog 函数现已更改为 File Dialog Express VI。双击该 VI 打开 **Selection Mode** 对话框，配置用户可以在文件对话框中选择文件或目录类型。
- Close File、Copy、Create Folder、Delete、File/Directory Info、Flush File 和 Move 函数包括新的接线器 (connector pane) 配置。新配置可影响由 LabVIEW 前期版本所创建的程序框图中的接线位置。
- 使用 Move 和 Copy 函数时，可以覆盖已有文件和目录。
- File/Directory Info 函数包括两个新输出端。**shortcut** 表示文件或目录是否为快捷方式，**resolved path** 返回快捷方式目标的路径。
- Read Key VI 的 String 实例不再包括 **multibyte encoding (0: None)** 参数，该参数在 LabVIEW 7.x 和前期版本中使用。

仪器 I/O 助手改进

仪器 I/O 助手 (Instrument I/O Assistant) 包括以下改进：

- 在 **Select Instrument** 步骤的 **Termination character** 下拉菜单中，可以选择字符用于表示仪器命令的最后一个字符。也可以选择 **<custom>** 在 **Termination character** 文本框中输入一个终止字符。
- 在 **Write** 步骤中，通过选择 **Enter a command** 文本框中的命令并单击 **Add parameter** 按钮，可以给仪器命令添加一个输入参数。也可以选择并右键单击该命令，从快捷菜单中选择 **Add parameter**。
- 可将 **Query and Parse** 步骤分为 **Write** 和 **Read and Parse** 两步以便于使用参数化的输入。要将 **Query and Parse** 步骤分步，右键单击 **Step Sequence** 窗口中的步骤并从快捷菜单中选择 **Split step into Write and Read steps**。
- 可以配置仪器 I/O 助手 (Instrument I/O Assistant)，分析所有数据直至完成所有分析。要将仪器 I/O 助手配置为分析所有数据，请在 **Query and Parse** 步骤中选择 **Count** 下拉列表的 **To end of data** 选项，或者单击 **Count** 下拉列表下的 **To end of data** 按钮。如单击 **Auto**

parse 按钮，仪器 I/O 助手也会分析所有数据。只有在配置最后一个 Token 时，才可使用 **To end of data** 选项和行为。

Sound VI 改进

Sound 选板中的 VI 可用于在 VI 中处理声音信号或录制声音。



注

(Windows) 使用 Sound VI 必须安装 DirectX 8.0 或更高版本。

这些 VI 包括以下改进：

- VI 可支持高达 24 位的声音信号。
- VI 可支持多通道声音文件以及单声道和立体声声音。
- 用波形数据表示声音数据。可以使用无符号 8 位整数、有符号 16 位或 32 位整数、单精度和双精度数据类型的元素来表示 Y 数组数据。每个波形均定义一条通道。
- 声音数据的格式是脉冲编码调制 (PCM)。
- VI 可以产生连续声音输出。
- VI 允许使用数据流的方式查看波形文件。
- VI 改进了错误检查功能。

字符串函数改进

使用标准 Perl Compatible Regular Expression (PCRE) 语法，Match Regular Expression 和 Search and Replace String 函数可以用于更复杂的字符串模式匹配。右键单击 Search and Replace String 函数并从快捷菜单中选择 **Regular Expression**，对函数的高级正则表达式搜索和子匹配替换字符串进行配置。调整 Match Regular Expression 函数的大小后，可以查看在字符串中找到的所有匹配子字符串。Match Pattern 函数或 Match Regular Expression 函数也可与用于搜索符合正则表达式的字符串。相对于 Match Pattern 函数，Match Regular Expression 函数可提供更多的字符串匹配选项，但也会相应降低执行性能。

VI 层次结构窗口改进

选择 **View»VI Hierarchy** 打开 **VI 层次结构 (VI Hierarchy)** 窗口，该窗口显示已打开的 LabVIEW 项目和终端以及内存中所有 VI 的调用层次结构。**VI Hierarchy** 窗口显示有一个顶层图标以表示主 LabVIEW 程序实例，该图标下面列出了所有不属于项目的已打开 VI。**VI Hierarchy** 窗口也将内存中的各个项目显示为一个顶层图标，所有被添加的终端均位于项目之下。

前面板改进

LabVIEW 8.0 包括以下前面板改进和修订。

图形游标改动

LabVIEW 8.0 包括以下图形游标 (Graph Cursors) 改动:

- 如需创建游标, 右键单击游标图例中任意区域并选择 **Create Cursor**, 然后从快捷菜单中选择游标模式。游标包括下列模式:

表 4 LabVIEW 8.0 游标模式改动

LabVIEW 7.1 游标模式	LabVIEW 8.0 游标模式	LabVIEW 8.0 游标模式说明
自由 (Free)	自由 (Free)	游标可在绘图范围内自由移动, 与曲线位置无关。
Snap to Point 和 Lock to Plot	单曲线 (Single-Plot)	游标只位于与游标相关的曲线上, 并可以在相关曲线中移动。右键单击游标图例行并从快捷菜单中选择 Snap To , 游标可与一条或所有曲线进行关联。
—	多曲线 (Multi-Plot)	游标只位于绘图范围内的特定数据点。多曲线游标可报告所有关联曲线的特定 X 值。可以将游标放在绘图范围内的任意曲线上。右键单击游标图例行并从快捷菜单中选择 Snap To , 游标可与一条或所有曲线进行关联。该模式只对混合信号图形有效。

- 创建游标模式后就不能对它进行修改, 除非删除游标并创建另一游标。右键单击游标图例行, 并从快捷菜单中选择 **Delete Cursor** 来删除某游标。
- 游标图例的右侧不再包括 **Cursor Movement Selector**、**Formatting Ring** 和 **Lock Ring** 按钮。右键单击游标图例所在行, 并从快捷菜单中选择相应选项以自定义游标。LabVIEW 前期版本中所有可访问的游标图例按钮现已添加到快捷菜单中。
- 利用 X Scale 属性可通过编程设置游标的 X 轴。该属性只对 **Free** 游标模式有效。该属性与游标图例快捷菜单中的 **X Scale** 选项类似。
- 利用 Y Scale 属性可通过编程设置游标的 Y 轴。该属性只对 **Free** 游标模式有效。该属性与游标图例快捷菜单中的 **Y Scale** 选项类似。
- 使用 Watch Plots 和 Watch All Plots 属性可通过编程查看绘图区域的标绘图 / 曲线。上述属性仅对与多曲线相关联的游标有效。

使用图形注释

图形注释 (Graph Annotation) 可用于在绘图区域内高亮显示数据点。注释包括一个标签和箭头, 用于指示注释内容和数据点。也可以配置注释外观以及注释在绘图区域的位置。如需创建注释, 右键单击图形并从快捷菜单中选择 **Data Operations»Create Annotation**, 打开 **Create Annotation** 对话框。右键单击注释, 并从快捷菜单中选择相应选项可以配置该注释。使用 Annotation List 属性可通过编程获取包括绘图区域内所有注释信息的数组。

在图形绘图范围内绘制图像

在图形的背景、前景或中间绘图范围 (Graph Plot Area) 中均可以绘制图像。波形图、强度图或混合信号图中的 Plot Images:Back、Plot Images:Front 和 Plot Images:Middle 属性可用于分别设置绘图范围的背景、前景和中间绘图图像。在强度图中仅可使用 Plot Image:Front 属性。绘图数据 (plot data) 位于前景和中间绘图图像之间。网格线 (grid line) 位于中间绘图图像和背景之间。如需从背景、前景或中间绘图范围删除现有图像, 必须将一个空图像连接至相应属性或将图像设置为透明 (transparent)。



注

如果调整图形大小, 绘图范围图像并不会进行相应调整, 必须返回到 VI 并重画该图像。

必须将图形坐标与前面板坐标进行映射, 才可以在图形绘图范围内绘制正确的图像。Map XY to Coordinates 方法可用于将图形或图表坐标与前面板坐标进行映射。该方法仅调整标绘图 / 曲线的左边界和上边界, 以便与前面板的原始图像相匹配。

关于使用 Plot Image 属性和 Map XY to Coordinates 方法的范例信息见 examples\general\graph\Graph Pictures.llb。

导出图形、图表和表格图像

图形 (graphs)、图表 (charts)、表格 (tables)、数字数据 (digital data)、数字波形输入控件和显示控件 (digital waveform controls and indicators) 的图像可导出至以下格式的文件:

(Windows) .emf、.bmp 和 .eps 文件

(Mac OS) .pict、.bmp 和 .eps 文件

(Linux) .bmp 和 .eps 文件

导出图像时, 右键单击控件并从快捷菜单中选择 **Data Operations» Export Simplified Image**。在 **Export Simplified Image** 对话框中, 选择图像格式并确定是否将图像保存至剪贴板或磁盘。也可利用 ExportImage 方法通过编程导出图像。

在 XY 图形中显示平面

在 XY 图形中可以显示 Nyquist 平面 (Plane)、Nichols 平面、S 平面和 Z 平面。如需显示上述平面, 右键单击该 XY 图形并从快捷菜单中选择 **Optional Plane**。Optional Plane 菜单选项列出了可在 XY 图形上显示的各种平面类型 (Nyquist、Nichols、S 和 Z 平面)。选择一种可显示的平面后, **Optional Plane** 菜单将包括各种菜单选项用于显示和隐藏平面线、平面标签和 Cartesian 线。

也可使用 **Optional Plane:Plane Type** 属性，通过编程在 XY 图形中显示上述平面。使用 **Optional Plane:Lines Visible** 和 **Optional Plane:Labels Visible** 属性可通过编程在 XY 图形中显示和隐藏平面线和平面标签。使用 **Cartesian Lines:Visible** 属性可通过编程在 XY 图形中显示或隐藏 Cartesian 线。

XY 图形也可显示复数的数组，其中 X 轴和 Y 轴分别显示实部和虚部。

混合信号图

混合信号图 (Mixed Signal Graph) 用于绘制可共享相同 x 轴的不同类型数据。混合信号图可接收所有波形图、XY 图形和数字波形图支持的数据类型。此外还可以接收由上述数据类型组成的任意簇。

将多个绘图 / 曲线添加至混合信号图时，各个绘图区域均包括单独的 y 轴。当需要添加模拟和数字数据组合时，混合信号图会自动创建绘图区域。

混合信号图的图例由树形输入控件组成，并显示在绘图范围的左侧。每个树形输入控件均表示一个绘图范围。使用图例可以在绘图范围中移动曲线。

在混合信号图中使用多曲线游标可以查看特定 x 点的多条曲线的值。

按键导航

右键单击前面板的控件并从快捷菜单中选择 **Advanced»Key Navigation**，可以显示 **Properties** 对话框的 **按键导航 (Key Navigation)** 选项卡。也可以右键单击控件，从快捷菜单中选择 **Properties** 并单击 **Key Navigation** 栏显示该选项卡。 **Key Navigation** 选项卡包含以下改动。

- **Key Assignment** 部分已更名为 **Focus**。 **Focus** 列表包含有以下快捷键： <Delete>、<Insert>、<Mute>、<Play>、<Volume Up>、<Volume Down>、F13、F14、F15、F16、F17、F18、F19、F20、F21、F22、F23 和 F24。
- **Current Assignments** 部分已更名为 **Existing Bindings**。
- **Key Navigation** 选项卡包含有 **Toggle** 部分，可用于指定拨动布尔控件 (toggle Boolean controls) 的快捷键。也可以使用 **Toggle Key Binding** 属性给拨动开关布尔控件分配快捷键。
- **Key Navigation** 选项卡现已包括 **Increment** 和 **Decrement** 部分，可指定快捷键用于增大或减小数字、滑动杆、旋钮、单选框、枚举等控件的值。也可以使用 **Increment Key Binding** 和 **Decrement Key Binding** 属性分别定义快捷键用于增大或减小数值控件的值。

Key Navigation 属性已更名为 **Focus Key Binding**。对于布尔控件，该属性仅用于设置 **focus** 键盘快捷键； **Toggle Key Binding** 属性用于设置 **toggle** 键盘快捷键。

最小化的前面板尺寸

在 LabVIEW 7.x 和前期版本中，前面板的最小尺寸以该面板的内容区域（不包括滚动条）为参考。在 LabVIEW 8.0 中，对于单面板 (single-pane) 的前面板，其最小尺寸也将以该面板的内容区域（不包括滚动条）为参考。但对于多面板 (multi-pane) 的前面板，其最小尺寸将参照整个前面板（包括任何可见的滚动条）。

分隔栏和面板

Containers 选板中的分隔栏 (Splitter Bars) 可将前面板分成两个或多个面板，并可单独滚动显示各个面板。通过这些面板可以设计带多个自定义区域的用户界面。利用分隔栏也可以将面板设计成一个工具栏或状态栏。

Pane 属性 (properties)、方法 (methods) 和事件 (events) 可用于通过编程方式控制面板。

Splitter 属性和事件可用于通过编程方式控制分隔栏。

LabVIEW 7.x 和前期版本使用的 **Customize Window Appearance** 对话框中的 **Show scroll bars** 选项在 LabVIEW 8.0 中被改成了两个选项：**Show horizontal scroll bar** 和 **Show vertical scroll bar**。如果在前面板中添加了一个分隔栏，那么这些选项将是不可修改的，同时这些选项只对左上角的面板有效。

自定义运行时快捷菜单

可自定义 VI 中任一控件的运行时快捷菜单 (Run-Time Shortcut Menu)。如需自定义该快捷菜单，右键单击控件并从快捷菜单中选择 **Advanced»Run-Time Shortcut Menu»Edit**，显示 **Shortcut Menu Editor** 对话框。使用该对话框可将默认快捷菜单或自定义快捷菜单文件 (.rtm) 与控件关联起来。使用 **Shortcut Menu Activation?** 事件和 **Insert Menu Items and Delete Menu Items** 函数可通过编程自定义快捷菜单。关于通过编程自定义控件的运行时快捷菜单的详细信息见 *LabVIEW Help*。

也可给前面板添加快捷菜单。要在前面板上添加快捷菜单，可使用 **Shortcut Menu Activation?** 和 **Shortcut Menu Selection** 事件。

如需禁用控件的运行时快捷菜单，也可以右键单击控件并从快捷菜单中选择 **Advanced»Run-Time Shortcut Menu»Disable**。

要禁用所有控件的默认快捷菜单，可选择 **File»VI Properties**，并从 **Category** 下拉菜单中选择 **Window Appearance**，单击 **Customize** 按钮并取消勾选 **Allow default run-time shortcut menus** 复选框。禁用默认运行时快捷菜单时，仍可以继续使用自定义快捷菜单。

使用 XControls

XControl 可用于设计和创建 LabVIEW 中复杂的输入控件和显示控件。

XControls 包括以下功能：

- 通过结合内置 LabVIEW 输入控件和显示控件创建 XControl，可充分利用 LabVIEW 输入控件和显示控件的功能。
- 在 VI 中使用 XControl 时，由于 XControl 包括控件的行为定义，从而可简化包含该 VI 的程序框图。
- 使用 XControl 库可确定该 XControl 的行为 (behavior)、外观 (appearance)、功能 (abilities)、属性 (properties) 和方法 (methods) 等。
- 可以自定义 XControl 的快捷菜单，使其包括所需新菜单选项和内置 LabVIEW 输入控件和显示控件的已有菜单选项。

在 **Project Explorer** 窗口中，右键单击 **My Computer** 结点并从快捷菜单中选择 **New»Control**，可以创建一个 XControl。



注

只可在 LabVIEW 专业版开发系统中创建和修改 XControl。如果 VI 包含 XControl，在所有 LabVIEW 软件版本中都可以运行该 VI。

关于 XControl 的范例信息见 `labview\examples\general\xcontrols\Dual Mode Thermometer` 目录中的 `Simple Dual Mode Thermometer XControl.lvproj`。

使用 .NET 控件

如 ActiveX 控件一样，可以在前面板中使用 .NET 控件。在 .NET container 中创建新控件或者将 .NET 控件添加到控件选板以便以后使用。在程序框图中，将 .NET 控件连接到 Property Node、Invoke Node 或 Register Event Callback 结点可设置属性、调用方法或处理控件事件。

创建新 .NET 控件，可将 .NET container 拖放到前面板中。右键单击该 .NET container 并从快捷菜单中选择 **Insert .NET Control**，显示 **Select .NET Control** 对话框。选择需创建的控件并单击 **OK** 按钮。

要将 .NET 控件添加至**控件选板**，请选择 **Tools»NET & ActiveX»Add .NET Controls to Palette**，显示 **Add .NET Controls to Palette** 对话框。选择需添加的控件并单击 **OK** 按钮。默认状态下，LabVIEW 将这些控件保存在 `labview\menus\Controls\DotNet & ActiveX` 目录中，因为该目录中的所有文件会自动出现在 **.NET & ActiveX** 选板中。

列表框、树形控件和表格

LabVIEW 8.0 提供各种扩展功能，包括用于列表框、表格、数字数据控件和树形控件的新属性和方法。上述四种控件可共享某些行为和功能。

- 右键单击多列列表框 (multicolumn listbox) 或树形控件 (tree control)，并从快捷菜单中选择 **Visible Items»Row Headers** 以显示

行首。也可以使用多列列表框或树形控件的 **Visible Items:Row Headers Visible** 属性，通过编程显示或隐藏行首。

- 右键单击多列列表框、表格或树形控件，并从快捷菜单中选择 **Visible Items»Vertical Lines** 或者 **Visible Items»Horizontal Lines** 以显示垂直和水平线。也可以使用多列列表框、表格或树形控件的 **Visible Items:Vertical Lines Visible** 属性，通过编程显示或隐藏垂直线。可以使用单列列表框 (single-column listbox)、多列列表框、表格或树形控件的 **Visible Items:Horizontal Lines Visible** 属性，通过编程显示水平线。
- 使用颜色工具可以给所有单元格、行首、列首或表格左上角单元格、列表框和树形控件单独上色。
- 如果文本超出单元格，在运行 VI 时如果将操作工具 (Operating tool) 的光标置于该单元格上，会出现完整文本的提示。
- 列表框和树形控件包括 4 个新条目符号。根据所使用的操作系统类型的不同，该复选框符号外观可能有所不同。右键单击控件中的条目并从快捷菜单中选择 **Item Symbol** 即可查看所有符号。列表框现可提供与树形控件相同的自定义符号功能。
- 在列表框和树形控件中，按下 <Shift-Enter> 键可在当前条目下插入一个新条目。在树形控件中，该按键组合所添加的新条目与当前条目处于同一层，并且所有位于新条目以下的条目都会下移一行。
- 右键单击列表框或树形控件并从快捷菜单中选择 **Editable Cells** 可允许用户在运行时编辑单元格。也可以使用 **Allow Editing Cells** 属性，通过编程来允许或禁止用户在 VI 运行时是否可以在单列列表框、多列列表框或树形控件中编辑单元格。使用 **Edit Cell** 事件可处理用户在单列列表框、多列列表框或树形控件中进行单元格文本修改的事件。
- 右键单击列表框或树形控件，并从快捷菜单中选择 **Highlight Entire Row** 以便在选择某项时高亮显示整行。
- 列表框和树形控件中键盘选择等行为与操作系统设置相同。



注

如在 **Properties** 对话框的 **Key Navigation** 选项卡中将 <PageUp>、<PageDown>、<Home> 或 <End> 指定用于某控件，则在列表框和树形控件中无法使用上述按键选择条目。

列表框

LabVIEW 8.0 包含以下列表框改进和修订：

- 右键单击单列列表框，并从快捷菜单中选择 **Visible Items»Horizontal Scrollbar** 可显示水平滚动条。也可以使用 **Visible Items:Horizontal Scrollbar Visible** 属性通过编程显示或隐藏水平滚动条。单列列表框快捷菜单的 **Visible Items»Scrollbar** 选项现已更名为 **Vertical Scrollbar**。**Scrollbar Visible** 属性现已更名为 **Visible Items:Vertical Scrollbar Visible**。

- 右键单击单列列表框，并从快捷菜单中选择 **Visible Items»Column Header** 可显示列首。也可以使用 **Visible Items:Column Header Visible** 属性通过编程显示或隐藏列首。
- 右键单击单列列表框中的行并从快捷菜单中选择 **Insert Row Before** 可在所选行之前添加新行。
- 右键单击单列列表框并从快捷菜单中选择 **Autosize Row Height**，允许各行自动改变高度以适应字体改变和多行文本输入等问题。也可以使用 **Autosizing Row Height** 属性通过编程允许各行自动调整高度。
- 右键单击单列列表框并从快捷菜单中选择 **Multi-line Input** 可允许单元格输入中包括多行文本。也可以使用 **Multiple Line Input** 属性，通过编程允许单元格输入可包括多行文本。
- 右键单击多列列表框并从快捷菜单中选择 **Smooth Scrolling** 可允许用户水平滚动列表框时，每次只滚动一个像素。也可以使用 **Smooth Horizontal Scrolling** 属性以便通过编程实现上述功能。

树形控件

LabVIEW 8.0 包含以下树形控件改进和修订：

- 在树形控件中，可以使用数字小键盘的+和-键打开和关闭条目。也可以使用右键单击该树形控件并从快捷菜单中选择 **Open All Items** 或 **Close All Items** 以打开或关闭所有条目。
- 在默认情况下，LabVIEW 会将树形控件中多个条目的交互式选择限制在属于同一父目录的条目中。如需选择不同父目录中的条目，请右键单击树形控件并从快捷菜单中取消勾选 **Selection Mode»Limit Multiselect to Siblings** 复选框。也可以使用 **Sibling Multiselect** 属性，通过编程将多个条目选择限制在同一目录中或允许选择不同父目录中的条目。



注

如果在树形控件中启用 **Limit Multiselect to Siblings** 快捷菜单选项，当选择多个不属于同一目录的条目时，后来的选项行为可能不会很直观。如果将选项重设为有效选项，该选项的行为将恢复正常。

- 右键单击树形控件并从快捷菜单中取消勾选 **Expand/Contract Symbol Type»Show at Indent Level 0** 复选框，可隐藏最高层次中条目的扩展 / 折叠 (expand/contract) 符号。缩进等级为 0 的条目将在树形控件的左侧展开。如果取消勾选该选项后的复选框，通过双击该条目仍可以扩展或折叠缩进等级为 0 的条目。要禁用双击功能，可将 **True** 常量连接至 **Item Close?** 事件的 **Discard?** 参数。也可以使用 **Expand/Contract Symbol:Show at Indent Level 0** 属性，通过编程显示或隐藏缩进层次 0 的扩展 / 折叠符号。

表格

如表格的水平滚动条不可见，表示运行时无法使用 <Shift- 右箭头> 键向右滚动。同样如果垂直滚动条不可见，表示无法使用 <Shift- 下箭头> 键向下

滚动。无论滚动条是否可见，都可以使用按键导航 (key navigation) 向上或向左滚动。

字符串和树形控件的拖放功能

在 LabVIEW 8.0 中，字符串和树形控件可支持内置的拖放功能。对字符串控件进行拖放操作时，利用操作 (Operating) 工具选择文本并将其拖放至可支持该文本的其它控件中，如其它字符串控件或树形控件。对树形控件进行拖放操作时，利用操作工具选择一个条目并将其拖放至可支持树形数据的其它控件中，如其它树形控件或字符串控件。

通过快捷菜单选项或事件、属性和方法等可以对控件中拖放的对象进行设置。执行拖放操作时，Get Drag Drop Data 函数可以从 LabVIEW 返回所拖动数据。

在字符串输入控件和显示控件中禁用自动换行

要在字符串输入控件或显示控件中禁用自动换行 (Word Wrapping)，右键单击该字符串并取消勾选 **Enable Wrapping** 快捷菜单选项的复选框。禁用自动换行后，字符串只在分行符后换行。如需禁用自动换行，字符串必须使用正常显示 (Normal Display) 模式。禁用自动换行后，通过右键单击该字符串输入控件或显示控件，并从快捷菜单中选择 **Visible Items» Horizontal Scrollbar** 可显示水平滚动条。

也可以使用 Enable Wrapping 属性，通过编程禁用自动换行。利用 Horizontal Scrollbar Visible 属性可以通过编程显示水平滚动条。

路径控件

Path Properties 对话框的 **Browse Options** 选项卡将取代 **Browse Options** 对话框。右键单击该路径控件 (Path Control)，并从快捷菜单中选择 **Browse Options** 以显示 **Browse Options** 选项卡。该选项卡包括以下改进：

- **Button Text** 文本框可用于修改按钮标签。也可使用 Browse Options: Button Text 属性，通过编程修改该按钮标签。
- 勾选 **Treat LLBs as folders** 复选框可允许用户在 LLB 中选择文件。
- 以下选项将取代 **Selection Mode** 下拉菜单：**Files only**、**Folders only**、**Files or folders**、**Existing only**、**New only** 和 **New or existing**。

滚动条控件

水平和垂直滚动条控件 (Scroll Bar Control) 用于在可显示滚动数据的控件中添加自定义滚动条。通过使用操作工具单击或将滚动框 (scroll box) 拖放至新位置，单击递增或递减箭头或滚动框与箭头之间的空白区域，可以相应改变滚动条的值。

也可以使用 Doc Min、Doc Max、Increment、Page Size、Coerce Minimum Value 和 Coerce Maximum Value 属性，通过编程确定页面大小和数据范围限制的最大和最小值。

数组滚动条

通过右键单击前面板中的数组输入控件或数组显示控件，并从快捷菜单中选择 **Visible Items»Vertical Scrollbar** 或 **Visible Items»Horizontal Scrollbar**，来显示数组的垂直或水平滚动条 (Array Scroll Bar)。只有维数大于 1 的数组可同时包括垂直和水平滚动条。垂直和水平调整一维数组大小可分别添加一个垂直滚动条和水平滚动条。

也可以使用 Vertical Scrollbar Visible 属性和 Horizontal Scrollbar Visible 属性，通过编程显示垂直和水平滚动条。

其它改进和修订

LabVIEW 8.0 包括以下前面板的其它改进和修订：

- 右键单击图形或图表，并从快捷菜单中选择 **Advanced»Disable Plot Legend** 可禁用或启用在运行时对图例进行修改。也可以使用 Legend:Disable 树形通过编程在波形图表、波形图或混合信号图中禁用或启用图例。禁用图例时，将无法在运行时自定义图例在图形或图标中的显示方式。
- 在 LabVIEW 7.x 和前期版本中，默认情况下数字波形图形的 y 轴不再显示轴标签。在 LabVIEW 8.0 中，默认情况下 y 轴不再显示轴标签。选择显示轴标签时，其默认文字现已由 **Amplitude** 更改为 **Digital Plots**。
- 列表框、多列列表框、表格、数字数据和树形控件现可支持鼠标滚轮垂直滚动浏览。
- 右键单击需进行自定义的输入控件和显示控件中某部分，并从快捷菜单中选择 **Advanced»Customize**，**Control Editor** 窗口中将只显示选中的要修改的控件。
- 移动滑块 (slider)、滑动杆指针 (needle of slide) 或旋转型 (rotary) 控件时，LabVIEW 会在提示框中显示该控件的值。提示框使用与该控件的数字显示框 (digital display) 相同的格式和精度。如该控件包括多个数字显示框，提示框所使用的格式和精度与当前指针和滑块相对应的数字显示框相同。如需禁用提示框，请右键单击控件并从快捷菜单中选择 **Properties**，单击 **Appearance** 栏并取消勾选 **Show value tip strip** 复选框。使用 Show Value Tip Strip 属性也可通过编程禁用提示框值。
- 数值输入控件和显示控件允许使用以下 SI 前缀：c、d、da 和 h。

LabVIEW MathScript（仅适用于 Windows）

LabVIEW MathScript 窗口 可用于编辑和执行数学命令、创建数学脚本以及查看变量的数字和图形化表示。选择 **Tools»MathScript Window** 打开

窗口。**LabVIEW MathScript 窗口**可产生输出并保持用户所调用命令的历史信息，同时可以列出已定义的变量以及显示所选择的变量。



注 只可在 LabVIEW 专业版开发系统中使用 MathScript。

也可以写入各种函数和脚本以便在 **LabVIEW MathScript 窗口**或 MathScript Node 中使用。

MathScript 节点 (MathScript Node) 可用于在程序框图中计算数学公式和表达式。可以保存在 **LabVIEW MathScript 窗口**中创建的脚本并加载至 MathScript Node。也可以保存在 MathScript Node 中创建的脚本并加载至 **LabVIEW MathScript 窗口**。

矩阵数据类型

矩阵数据类型 (Matrix Data Type) 可使数学问题建模更简单有效。现有用于矩阵操作的多态 VI 和函数可以识别矩阵数据类型并实现相关的矩阵算法。用于复数矩阵操作的数学 (Mathematics) VI 和函数可将矩阵作为输入，并在结果中产生相应矩阵。



注 在 LabVIEW 7.x 和前期版本中，Matrix Exp、Matrix Logarithm、Matrix Power 和 Matrix Square Root VI 仅可用于 LabVIEW 完整版和专业版开发系统。在 LabVIEW 8.0 中，基础版 (Base Package) 中也已包含上述 VI。

由于矩阵数据可存储包括实数或复数标量数据的行或列，以便在矩阵操作中使用 (尤其是部分线性代数操作)，因此矩阵数据类型可替代二维数组用于表示矩阵数据。执行矩阵操作的数学 VI (Mathematics VI) 和函数接收矩阵数据类型并返回矩阵结果，这样数据流中下面的多态 VI 和函数就可以执行矩阵特有的操作。如果不执行矩阵操作的 Mathematics VI 或函数可支持矩阵数据类型，该 VI 或函数会自动将矩阵数据类型转换为二维数组。如果将二维数组连接至可默认执行矩阵操作的 VI 或函数，根据二维数组的数据类型，该 VI 或函数会自动将二维数组转换为实数或复数矩阵。

大多数数值 (Numeric) 函数支持矩阵数据类型和矩阵操作。例如，Multiply 函数可用于将一个矩阵与另一个矩阵或数字相乘。通过基本数值数据类型和复数线性代数函数，可创建用于执行精确矩阵操作的数值算法。

在 LabVIEW 7.x 和前期版本中，需使用 Linear Algebra VI 在矩阵中执行数学操作。尽管二维数组可以存储矩阵数据，对于某些关键数值函数 (如部分线性代数函数)，LabVIEW 处理二维数组的方式与其处理矩阵控件中的矩阵数组的方式并不相同。LabVIEW 7.x 和前期版本无法区分二维数组与矩阵，因此在执行数学函数之前必须确认数组的维数和数组元素，从而确保用于该矩阵操作的矩阵数组的正确性。如果使用矩阵数据类型，在执行数值函数前无需检查数组的维数或分析数组元素。如果所连接的两个矩阵的维数不同，例如两个矩阵分别为 2×3 和 4×4 矩阵，该 VI 或函数将返回一个空矩阵。



注

VI 或函数将数据在矩阵和二维数组之间进行转换时，该 VI 或函数上将出现强制转换点 (Coercion dots)。由于 LabVIEW 保存矩阵和二维数组的方式相同，该类数据转换并不会影响整体性能。

在程序框图中，矩阵数据类型外观与实数二维数组或复数二维数组类似，但它们的接线模式 (wire pattern) 不同。在可支持矩阵数据类型的 VI 或函数中连接一个矩阵数据类型作为其输入，该 VI 或函数将自动支持与矩阵相关的各种操作。

将矩阵数据类型作为输入连接至以下函数时，包含用于处理矩阵数据类型子 VI 的 VI 将取代该函数：

- Equal? 函数
- Not Equal? 函数
- Absolute Value 函数
- Add 函数
- Multiply 函数
- Square Root 函数
- Subtract 函数
- Exponential 函数
- Natural Logarithm 函数
- Power Of X 函数
- Re/Im To Complex 函数
- Complex To Re/Im 函数
- Polar To Complex 函数
- Complex To Polar 函数

所得到 VI 的图标相同，但其中已包括与矩阵相关的算法。如果在输入端断开与矩阵的连接，该节点使用的 VI 不会改变。而连接其它数据类型作为输入时，该节点将恢复为原来的函数。如果将一个数据类型与函数连接，并且该数据类型可引起基本数学操作错误，该函数将返回一个空矩阵或 **NaN**。例如，如果将维数是 2×3 的矩阵连接至 **Multiply** 函数的一个输入，并将维数是 4×4 的矩阵连接至另一输入，该函数将返回一个空矩阵。



注

将 LabVIEW 7.x 的 VI 加载至 LabVIEW 8.0 时，并且该 VI 的 Mathematics VI 所连接的函数可以使用矩阵数据类型，LabVIEW 将禁用自定义类型并保持 LabVIEW 7.x 的行为。LabVIEW 会在该函数上标注红色的 7.x 符号，以便表示该函数使用 LabVIEW 7.x 的行为。

在 LabVIEW 7.0 和前期版本中，内存中只有一维数组已进行对齐，这有利于提高数组操作的性能。在 LabVIEW 7.1 和更高版本中，内存中的一维和二维数组均已进行对齐，从而提高了线性代数操作以及与矩阵数据类型相关的操作的性能。

选项对话框改进

选择 **Tools»Options** 打开 **Options** 对话框。选择对话框左侧的 **Category** 列表，可以浏览 **Options** 对话框的所有选项卡。**Options** 对话框包括以下改进：

- **Miscellaneous** 选项卡已更名为 **Environment** 选项卡。**Maximum undo steps per VI** 选项已从 **Block Diagram** 选项卡移至 **Environment** 选项卡。
- **VI Server:TCP/IP Access** 选项卡已更名为 **VI Server:Machine Access** 选项卡并已重命名部分选项。
- **VI Server:User Access** 选项卡可用于控制用户从 VI Server 访问 VI 的权限。
- **Menu Shortcuts** 选项卡可用于设置 LabVIEW 菜单选项的键盘快捷键。
- **Source Control** 选项卡(仅支持 LabVIEW 专业版开发系统)可用于配置第三方源代码控制软件的源代码控制 (source control) 并在 LabVIEW 中设置源代码控制选项。
- **Security** 选项卡可用于设置 LabVIEW 的安全选项。
- **Shared Variable Engine** 选项卡可用于设置共享变量的时间同步。
- **Front Panel** 和 **Block Diagram** 选项卡的 **Labels snap to preset positions on controls/terminals**、**Labels locked by default** 和 **Default label position** 选项可用于抓取、锁定或定位新对象的标签和标题。
- **Controls/Functions Palettes** 选项卡中 **Format** 下拉菜单的 **Tree** 选项可用于在树形控件中显示选板项目。如果从 **Format** 下拉菜单选择 **Text** 或 **Tree** 选项，并勾选 **Sort palette items** 复选框，将按字母顺序对选板上的条目进行排序。
- **VI Server: Configuration** 选项卡的 **Service name** 选项可用于设置 VI Server 监听请求的 TCP/IP 服务名称。

另存为和保存为前期版本对话框

在早期已保存 VI 中，选择 **File»Save As** 打开 **另存为 (Save As)** 对话框。选择 **File»Save for Previous Version** 可打开新的 **保存为前期版本 (Save for Previous Version)** 对话框。

Save with Options 对话框已删除，现可使用 **Save As** 对话框或 **Save for Previous Version** 对话框。也可以从 **Save As** 对话框中查看 **Source Distribution Properties** 对话框，可创建一个源代码发布 (source distribution) 并配置特定 VI 从而添加密码、删除程序框图或应用其它设置等。

文件大小改进

在 LabVIEW 8.0 中保存 LabVIEW 前期版本的 VI 可有效减小文件大小。相对于 LabVIEW 7.1 所保存的 VI，LabVIEW 8.0 保存相同 VI 的文件体积大

约可减少 55%。相对于 LabVIEW 7.1 所保存的 LLB，LabVIEW 8.0 保存相同 LLB 的文件体积大约可减少 20%。

在各个版本的 LabVIEW 中使用动态链接库

在 LabVIEW 7.x 和前期版本中，如果使用某个版本 LabVIEW 创建一个动态链接库 (DLL)，并在其它版本 LabVIEW 中打开该动态链接库，由于调用该动态链接库的 LabVIEW 运行引擎 (Run-Time Engine) 版本不同，LabVIEW 可能会返回错误信息。在 LabVIEW 8.0 中，如果使用 `labvieww.lib` 而不是 `labview.lib` 创建动态链接库，可以在不同版本的 LabVIEW 或运行引擎中使用该动态链接库。

设置窗口运行位置

选择 **File»VI Properties**，并从 **Category** 下拉菜单中选择 **Window Run-Time Position**，从而自定义前面板运行窗口的位置和大小。也可以使用 `Front Panel:Run-Time Position` 方法通过编程自定义前面板运行窗口的位置和大小。

Position 下拉菜单中的 **Maximized** 选项将取代 **Size the front panel to the width and height of the entire screen** 复选框，后者位于 LabVIEW 前期版本的 **VI Properties** 对话框的 **Window Size** 选项卡上。

同时 **Position** 下拉菜单中的 **Centered** 选项也将取代 **Auto-Center** 复选框，后者位于 LabVIEW 前期版本的 **Customize Window Appearance** 对话框中。

在 LabVIEW 前期版本中，如果将 VI 设置为 **Auto-Center** 或 **Size the front panel to the width and height of the entire screen**，停止运行时 VI 无法返回至编辑模式位置。VI 将保持居中或整个屏幕大小。LabVIEW 8.0 在停止时 VI 可返回至编辑模式位置。

设置运行语言选项

通过以下方式可以为独立应用程序设置语言选项。以下方法按从高到低的优先级顺序进行排列。

- 运行包括 `-lang language` 命令行选项的应用程序，其中 `language` 是所选语言的英语名称。
- 将以下行添加至该应用程序的 `.ini` 文件：`AppLanguage=language`，其中 `language` 是所选语言的英语名称。
- **(Windows)** 如果没有在命令行或应用程序的 `.ini` 文件中指定所需语言，该应用程序将使用与操作系统相同的语言。
- 如果应用程序无法支持操作系统使用的语言，将使用应用程序的默认语言。
- 如果没有指定上述任一语言选项，应用程序将按字母顺序使用第一种可支持的语言。如果应用程序无法使用任一可支持的语言，LabVIEW 将无法加载该应用程序。

VI Server 改进

相对于 LabVIEW 7.x 和前期版本，VI Server 提供了更详细的错误报告。例如，使用 **Open VI Reference** 函数发生错误时，错误报告会列出导致错误的 VI。如果加载的 VI 无前面板，错误信息中将说明 LabVIEW 无法加载前面板。错误报告也包括产生错误的特定属性和方法的名称。

使用 VI Server 引用

控件引用 (control reference) 常量已更名为 VI Server 引用。将该引用置于程序框图中，并链接至当前 VI、应用程序或者 VI 的输入控件或显示控件。通过该引用可访问与其链接的 VI、应用程序、输入控件或显示控件的属性和方法。

使用操作工具单击程序框图中的 VI Server 引用，并选择 **This Application** 或 **This VI** 将引用链接至当前应用程序或 VI。也可以右键单击该 VI Server 引用，并从快捷菜单中选择 **Link to»This Application** 或 **Link to»This VI**。要将该引用链接至 VI 中的输入控件或显示控件，以同样的方式单击或右键单击该引用，并从快捷菜单中选择需链接的输入控件或显示控件。

由 LabVIEW 创建的可执行程序无法打开到多态 VI 的 VI Server 引用。当 LabVIEW 创建可执行程序时，LabVIEW 将删除多态 VI 并在可执行程序中创建相应多态实例。可以直接对多态实例执行 VI Server 操作。

用于 VI Server 的 NI Security

VI Server 中包括的 NI Security 可用于更好地控制哪些客户端可以连接至 VI Server。在 LabVIEW 7.x 和前期版本中，VI Server 可以根据客户端的机器地址允许或禁止对服务器的连接。LabVIEW 8.0 已经集成了 NI Security，VI Server 可以根据用户或组允许或禁止对服务器的连接。

默认情况下，VI Server 已禁用 NI Security。在禁用状态下，用户仅可以根据机器地址允许或禁止连接。启用 NI Security，必须指定允许或禁止连接服务器的用户和组。选择 **Tools»Options** 打开 **Options** 对话框，并从 **Category** 列表中选择 **VI Server: User Access** 打开 **VI Server: User Access** 选项卡，从而设置允许或禁止访问权限。如果用户控制访问权限列表为空，则表示 VI Server 已禁用 NI Security。

在 LabVIEW 8.0 中，客户端打开 VI Server 连接的方法与 LabVIEW 7.x 和前期版本相同。如果服务器已经设置了 VI Server 的用户或组限制，客户端可能需要先登录才可以连接服务器。使用 **NI Security Login** 对话框进行登录，或者利用 NI Security:Login 方法通过编程登录服务器。如果用户没有获得连接许可，服务器将关闭连接并返回

`kLVE_NISecurity_AuthenticationFailed (1379)` 错误信息。

如果在 VI Server 打开时改动客户端，该客户端将自动给服务器发送信息以表示用户已改变。如果新用户没有获得连接许可，服务器将关闭 VI Server 连接并返回 `kLVE_NISecurity_AuthenticationFailed (1379)` 错误信息。

类浏览器窗口

类浏览器窗口 (Class Browser Window) 用于选择可用对象库，并在所选择对象库中查看类、属性和方法。可以创建所选择属性或方法的属性节点 (Property Node) 或调用节点 (Invoke Node)。也可以通过 **Class Browser** 窗口搜索类、属性和方法或者创建子属性和子方法。选择 **View»Class Browser** 可以打开 **Class Browser** 窗口。

要编辑现有属性或调用节点，可以右键单击属性节点的属性并从快捷菜单中选择 **Select Property**，或者右键单击调用节点并从快捷菜单中选择 **Select Method**。

使用 .NET 集合 (.NET Assembly)

LabVIEW 将自动加载 .NET 通用语言运行库 (Common Language Runtime)(CLR) 的最新安装版本。如果使用 .NET 前期版本开发 .NET 应用程序，可安装 .NET 的更高版本并重新发布该应用程序，而 .NET 前期版本用户将无法运行该程序。使用配置文件可以加载特定 CLR 版本。这样，用户可以安装 .NET 更高版本并继续开发 .NET 应用程序，同时其它用户可以继续在 .NET 前期版本中运行该程序。

在 LabVIEW 7.x 中，必须选择 **Tools»Advanced» .NET Assembly References** 进行手动注册专用 (private).NET 集合文件，以便运行引用上述集合 (assembly) 的 VI。在 LabVIEW 8.0 中无需注册专用 .NET 集合文件，LabVIEW 8.0 已不包括 **Tools»Advanced» .NET Assembly References** 菜单选项，现可利用构造器节点 (Constructor Node) 创建特定专用 .NET 集合的引用。LabVIEW 将保存该集合到包含 .NET 对象的 VI 的相关路径。如果在 LabVIEW 项目文件目录或子目录中的 Global Assembly Cache (GAC) 保存集合，则 LabVIEW 可更快地加载上述集合。

.NET 通用语言运行库 (CLR) 通过配置的设置来决定所加载的集合版本。该配置设置可能会使所请求的集合特定版本失效。例如，系统管理员可将计算机配置为加载 1.0.0.1 版而不是 1.0.0.0 版。如果试图加载 1.0.0.0 版，.NET CLR 将通知所返回的版本返回集合 1.0.0.1 版，同时 LabVIEW 也将显示一个对话框通知所返回的版本。

如果将使用专用集合的 VI 移至其它文件夹或计算机，必须确保该集合与该 VI 所保存的相对位置相同。如果 LabVIEW 无法在 GAC、包含项目的目录或 VI 所保存的相对路径中找到集合文件，LabVIEW 会提示您查找该集合文件。

如将使用专用集合的 VI 创建成动态链接库或独立应用程序，LabVIEW 会将相关专用 .NET 集合文件复制至该库或应用程序目录中的 data 子目录。

在 LabVIEW 8.0 中，现可将 .config 文件应用于已保存的项目、动态链接库或独立应用程序。

仪器驱动查找工具

仪器驱动查找工具 (Instrument Driver Finder) 可用于在 LabVIEW 开发环境中更简单地查找和安装 LabVIEW 即插即用仪器驱动程序，选择 **Tools»Instrumentation»Find Instrument Drivers** 即可打开该工具。仪器驱动查找工具也可以显示 `instr.lib` 中的仪器驱动程序，并利用 NI-VISA 搜索已连接的仪器。



注 在 Mac OS 中无法使用仪器驱动查找工具。

仪器驱动 VI 向导

仪器驱动 VI 向导 (Instrument Driver VI Wizard) 可以扩展现有仪器驱动项目库。创建或打开仪器驱动项目库后，在 **Project Explorer** 窗口中右键单击项目库，并从快捷菜单中选择 **New»Instrument Driver VI** 可以打开仪器驱动 VI 向导。通过该向导创建一个仪器驱动 VI，并将该 VI 添加到仪器驱动项目库中。该向导可以配置仪器命令字符串的参数并且定义仪器响应的方式，仪器命令字符串将与前面板控件相对应。

仪器驱动项目向导

仪器驱动项目向导 (Instrument Driver Project Wizard) 可以使仪器驱动项目的创建过程自动化，其中包括用于控制诸如 GPIB、RS-232、Ethernet 和 USB 仪器等可编程仪器的各种 VI，选择 **Tools»Instrumentation»Create Instrument Driver Project** 即可运行该向导。该向导将创建仪器驱动 VI 和用于在 **函数** 选板中查看驱动 VI 的选板菜单。一个类模板、基于消息的通用模板或类似仪器的现有驱动程序均可以用于创建一个仪器驱动程序。

现可使用的类模板包括 DC Power Supply、Digital Multimeter、Function Generator 和 Oscilloscope。通用模板可用于任何基于消息的仪器。如果仪器类似于某个现有仪器，可以根据现有仪器驱动程序创建新的仪器驱动。

如果根据现有仪器驱动程序创建新仪器驱动，打开仪器驱动项目向导之前必须先将现有仪器驱动项目安装至 `labview/instr.lib` 目录。现有仪器驱动程序必须是属于项目类型的驱动程序。通过 NI 仪器驱动查找工具即可下载和安装项目类型的驱动程序。

仪器驱动项目向导将创建一个 LabVIEW 项目，并且该项目将使用 LabVIEW 即插即用仪器驱动的标准格式。项目中包括了符合仪器驱动程序指导方针的 VI 和选板菜单文件。向导创建项目后，必须自定义各个 VI 以便与仪器正常工作。通过该向导可以创建一个仪器驱动 VI，并将该 VI 添加到仪器驱动项目库中。

使用和调试可重入 VI

通过可重入的 VI (Reentrant VI)，LabVIEW 可完成以下任务：

- 可以对可重入的 VI 进行调试。要调试可重入的 VI，可以查看可重入 VI 的程序框图实例但不能对程序框图实例进行编辑。在程序框图中，可以设置断点 (breakpoint)、使用探针 (probe)、启用执行高亮显示 (execution highlighting) 和单步 (single-step) 执行。如需调试某个 Reentrant VI，可选择 **File»VI Properties** 以显示 **VI Properties** 对话框，从下拉菜单中选择 **Execution** 并勾选 **Allow debugging** 复选框。
- 通过 **VI Properties** 对话框可将重入 VI 设置为执行过程中打开前面板并在该 VI 完成运行后可选择关闭前面板。
- 在运行时控制可重入 VI 的前面板的输入控件和显示控件，现可使用 VI Server 通过编程实现；但在运行时无法编辑输入控件和显示控件。如果引用某个 VI 的重入实例 (reentrant instance)，可以使用多种 VI Server 方法和属性与该实例进行通信，如 Control Value:Get (Flattened)、Control Value:Set (Flattened) 和 Run VI 等。
- 也可以使用 VI Server 创建运行时 Reentrant VI 的前面板副本。如需复制可重入 VI 的前面板，可使用 Open VI Reference 函数打开一个 VI Server 引用。将一个严格类型的 VI 引用连接至 **type specifier** 输入端或者将 **open for reentrant run** 与 **option** 输入端连接。打开一个引用时，LabVIEW 将创建该 VI 的一个副本。也可以使用 VI Server 或 **VI Properties** 对话框打开 reentrant VI 的前面板。
- 通过配置事件结构 (Event structure) 来处理一个重入 VI 的前面板对象的事件。
- 通过与 LabVIEW 内置 Web 服务器连接，可以在 LabVIEW 内或 Web 浏览器中远程查看和控制重入 VI 的前面板。使用 Web 服务器时，可允许多个用户访问某个 VI 的控制器界面。每个连接均可获取各自的实例。
- XControl 的 Facade VI 必须是一个可重入 VI。
- 可重入 VI 的前面板也可以是子面板。
- 在 LabVIEW 前期版本中，重入 VI 的每个实例仅可显示一个前面板，因此无法保证输入控件和显示控件值的准确性。在 LabVIEW 8.0 中，每个重入 VI 的各个实例均包括单独的前面板，以确保每次输入控件和显示控件值的准确性。

从程序框图中打开一个可重入子 VI 时，LabVIEW 将打开该 VI 的一个副本而不是 VI 本身。VI 的标题栏中包括 (clone) 以表示它是源 VI 的一个副本。无法对 VI 副本进行编辑。

在 VI 运行后创建探针

将 VI 配置成保持连线值后，在程序框图中创建一个探针时，该探针将显示 VI 最后执行时流经该连线的数值。通过单击 **Retain Wire Values** 按钮，可以使 VI 保存执行过程中各个点的连线值。这样，只要在连线上放置一个

探针，用户就可以马上获取该连线的最新数据值。使用探针从任一连线获取数据之前，必须至少成功运行 VI 一次。当程序框图很复杂并且需要调试 VI 时，在 VI 运行后使用探针将非常有用。



单击程序框图工具栏中的 **Retain Wire Values** 按钮可以保持连线值以便在探针中使用。



在任意时间单击程序框图工具栏中的 **Do Not Retain Wire Values** 按钮可以避免在每次执行时保存连线值。



注

选择 **Do Not Retain Wire Values** 选项可以减少内存需求并提高系统性能。

记录多个终端中的 VI 信息

通过 **Profile Performance and Memory** 窗口可以同时记录不同终端上的 VI 信息。**Profile Performance and Memory** 窗口包括一个新的 **Project Library** 列，可显示 LabVIEW 项目库 (*.lvlib) 和终端。该列可用于区分位于不同项目库或加载至不同终端中的同名 VI。**Profile Performance and Memory** 窗口包括一个新的 **Select Application Instances** 按钮，单击按钮可打开 **Select Application Instances** 对话框。通过该对话框可选择在 **Profile Performance and Memory** 窗口中显示哪些终端的数据。**Select Application Instances** 按钮旁边是一个图例，用于显示通过 **Profile Performance and Memory** 窗口访问的各个终端以及 LabVIEW 用来表示各个终端的颜色。选择 **Tools»Profile»Performance and Memory** 可显示 **Profile Performance and Memory** 窗口，而在前期版本中应选择 **Tools»Advanced»Profile VIs**。



注

LabVIEW 项目 (*.lvproj) 是一个包括 VI、程序生成规范等的集合。所有与项目相关的文件将出现在 **Project Explorer** 窗口中。每次只能记录一个项目的信息。如果已经打开 **Profile Performance and Memory** 窗口，将禁用 **Tools»Profile»Performance and Memory** 选项。

64 位整数数据类型

通过 64 位有符号和无符号整数数据类型可查看、使用和保存较大的整数。

所有带多态数值输入的函数均可支持 64 位整数数据类型。LabVIEW 会将 Rotate、Logical Shift 和 Scale By Power Of 2 函数的 rotation amount、shift amount 和 power 的 64 位输入分别强制转换为 32 位整数。如需函数返回一个 64 位整数输出，必须将 64 位整数连接至 Decimal String To Number、Hexadecimal String To Number、Octal String To Number 和 Fract/Exp String To Number 函数的 **string** 输入端。

使用 To Quad Integer 和 To Unsigned Quad Integer 函数可分别将数字转换为 64 位有符号和无符号整数。

NI Spy

NI Spy 是在应用程序运行时监测仪器 I/O 通信的工具。在 LabVIEW 运行 VI 时可以使用 NI Spy 记录仪器 I/O 的调用和结果信息。记录调用和结果信息可帮助您调试仪器通信中所碰到的问题。**(Windows)** 选择 **Start»National Instruments»NI Spy** 运行 NI Spy。**(Mac OS)** 选择 **Applications»National Instruments»NI Spy** 并双击 NI Spy 图标运行 NI Spy。**(Linux)** 在命令行中输入 `nispy` 运行 NI Spy。

查找和替换功能

在 LabVIEW 7.x 和前期版本中替换程序框图中的对象，可以右键单击对象并从快捷菜单中选择 **Replace**。在 LabVIEW 8.0 中，您可以同时替换多个对象或文本。**Find** 对话框可用于搜索 VI，并找到所需 VI、对象和文本。然后使用 **Search Results** 窗口将所有或部分选项替换为其它对象或文本。

自定义键盘快捷键

Options 对话框的 **Menu Shortcuts** 选项卡可用于设置 VI 菜单选项的键盘快捷键。无需 <Ctrl> 就可以将 <F1> 到 <F24> 等功能键作为快捷键使用。

打印对话框改进

打印 (Print) 对话框包括以下改进：

- **Select VI(s)** 选项卡可用于选择只打印当前 VI 还是打印多个 VI。
- 通过 **Print Contents** 选项卡的 **Complete front panel**、**Visible portion of front panel** 和 **VI documentation** 选项可选择需打印的内容。**Using the panel**、**Using as a SubVI**、**Complete documentation** 和 **Custom** 文档选项已从 **Print Contents** 选项卡转移至 **VI Documentation** 选项卡，后者可通过选择 **VI documentation** 选项查看。
- **Custom details** 选项卡已更名为 **VI Documentation** 选项卡。**VI Documentation Style** 选项可用于设置 **VI Documentation** 选项卡所包含的组件。**Icon and connector pane** 和 **Description** 选项将取代 **Icon and description** 选项及其组件。**Controls** 选项下的 **Label**、**Caption** 和 **Caption (Label)** 组件可用于打印输入控件和显示控件的标签和 / 或标题 (caption)。
- 该对话框可打印当前程序实例中 VI 的文档。如需打印多个程序实例中 VI 的文档，必须在各个程序实例中重复上述打印操作。

编辑 VI 图标

利用 `labview\resource\plugins` 目录中的 `lv_icon` VI 模板可以创建一个 VI，无需使用 **Icon Editor** 对话框即可通过该 VI 编辑 VI 图标。关于创建图标编辑器 VI 的范例信息见 `SAMPLE_lv_icon` VI。

文档改进和修订

LabVIEW 8.0 包括以下文档修订和改进。

文档重排

LabVIEW Help 现在包括 LabVIEW 7.x 以及早期产品手册和应用说明中的概念内容，因此所发布的 LabVIEW 不再包括 PDF 库或 PDF 版本的 *LabVIEW 测量手册 (LabVIEW Measurements Manual)*、*在 LabVIEW 中使用外部代码 (Using External Code in LabVIEW)*、*LabVIEW 开发指南 (LabVIEW Development Guidelines)*、*LabVIEW 分析概念 (LabVIEW Analysis Concepts)* 或应用说明等。

下列表格说明 LabVIEW 7.x 和 LabVIEW 8.0 中文档的位置。

表 5 LabVIEW 8.0 文档重组

文档	LabVIEW 7.x	LabVIEW 8.0
《LabVIEW 入门指南》 (<i>Getting Started with LabVIEW</i>)	印刷品和 PDF	印刷品和 PDF
《LabVIEW 发行说明》 (<i>LabVIEW Release Notes</i>)	印刷品和 PDF	印刷品和 PDF
《LabVIEW 升级说明》 (<i>LabVIEW Upgrade Notes</i>)	印刷品和 PDF	印刷品和 PDF
《LabVIEW 快速参考指南》 (<i>LabVIEW Quick Reference Card</i>)	印刷品和 PDF	印刷品和 PDF
《LabVIEW 用户手册》 (<i>LabVIEW User Manual</i>)	印刷品和 PDF	<i>LabVIEW Help: LabVIEW 基础 (Fundamentals)</i> (包含于印刷品中)
<i>LabVIEW Measurements Manual</i>	印刷品和 PDF	<i>LabVIEW Help: Taking Measurements; Controlling Instruments</i>
<i>LabVIEW Application Builder Readme</i>	印刷品和 PDF (<i>LabVIEW Application Builder User Guide</i>)	labview\readme 目录
<i>LabVIEW Development Guidelines</i>	印刷品和 PDF	<i>LabVIEW Help: Fundamentals» Development Guidelines</i>
<i>LabVIEW Analysis Concepts</i>	PDF	<i>LabVIEW Help: Fundamentals» Signal Processing and Analysis</i>

表 5 LabVIEW 8.0 文档重组 (续)

文档	LabVIEW 7.x	LabVIEW 8.0
<i>Using External Code in LabVIEW</i>	PDF	<i>LabVIEW Help: Fundamentals» Calling Code Written in Text-Based Programming Languages</i>
<i>Integrating the Internet into Your Measurement System-DataSocket Technical Overview</i>	PDF	—
<i>LabVIEW and Hyper-Threading</i>	PDF	<i>LabVIEW Help: Fundamentals» LabVIEW and Hyperthreading</i>
<i>LabVIEW Custom Controls, Indicators, and Type Definitions</i>	PDF	<i>LabVIEW Help: Fundamentals» Building the Front Panel» Concepts»Creating Custom Controls, Indicators, and Type Definitions</i>
<i>LabVIEW Data Storage</i>	PDF	<i>LabVIEW Help: How LabVIEW Stores Data in Memory</i>
<i>LabVIEW Performance and Memory Management</i>	PDF	<i>LabVIEW Help: Fundamentals» Managing Performance and Memory»Concepts»Using the VI Profile Window to Monitor VI Performance</i>
<i>Polymorphic Units in LabVIEW</i>	PDF	<i>LabVIEW Help: Fundamentals» Creating VIs and SubVIs» Concepts»Using Polymorphic Units</i>
<i>Porting and Localizing LabVIEW VIs</i>	PDF	<i>LabVIEW Help: Fundamentals» Sorting and Localizing VIs</i>
<i>Using Apple Events and the PPC Toolbox to Communicate with LabVIEW Applications on the Macintosh</i>	PDF	—
<i>Using LabVIEW to Create Multithreaded VIs for Maximum Performance and Reliability</i>	PDF	<i>LabVIEW Help: Fundamentals» Definitions:Multitasking, Multithreading, and Multiprocessing</i>
<i>Using LabVIEW with TCP/IP and UDP</i>	PDF	<i>LabVIEW Help: Using LabVIEW with TCP/IP and UDP</i>

表 5 LabVIEW 8.0 文档重组（续）

文档	LabVIEW 7.x	LabVIEW 8.0
<i>Using LabVIEW with Wireless Devices</i>	PDF	<i>LabVIEW Help: Using LabVIEW with Wireless Devices</i>
<i>Using the Timed Loop to Write Multirate Applications in LabVIEW</i>	PDF	<i>LabVIEW Help: Fundamentals» Loops and Structures» Timed Structures</i>

(Windows) 请按以下步骤打印 *LabVIEW Help* 中 **目录** 栏中的所有主题内容：

1. 右键单击该标题。
2. 从快捷菜单中选择 **打印** 打开 **打印主题** 对话框。
3. 选择 **打印选定标题及全部次主题** 选项。



注 如希望只打印 **目录** 中所选单个主题，请选择 **打印选定主题**。

4. 单击 **确定** 按钮。

该帮助文件可能包括 PDF 文档链接。如需打印 PDF 文档，单击 Adobe Acrobat 查看工具栏中的打印按钮。

使用 Adobe Reader with Search and Accessibility 6.x 或更高版本搜索 PDF 版本的 *LabVIEW User Manual*、*LabVIEW Release Notes*、*LabVIEW Upgrade Notes*、*Getting Started with LabVIEW Manual* 和 *LabVIEW Application Builder Readme*。请按以下步骤搜索所有 LabVIEW PDF 文档。

1. 选择 **Edit»Search**，显示 **Search PDF** 窗口。
2. 在 **What word or phrase would you like to search for** 文本框中输入单词或短语。
3. 单击 **All PDF Documents in** 按钮并在下拉菜单中选择 labview\manuals 目录。
4. 单击 **Search** 按钮。

关于在文件夹的所有 PDF 文档中搜索单词或短语的详细信息见 Adobe Reader 帮助文档。

自述文件

LabVIEW 文件结构中的 readme 目录包括 LabVIEW 自述文件和所有已安装模块和工具套件的自述文件。通过选择 **开始 » 程序 » National Instruments»LabVIEW»Readme** 可查看自述文件。

定位按钮

在 *LabVIEW Help* 中，如果使用**索引**、**搜索**或**书签**栏查找到某一主题，请单击**定位**按钮以便在目录中查找该主题。



注 在 LabVIEW 帮助中，目录栏下面未列出对话框相关主题。

查看范例

右键单击程序框图中或锁定选板上的 VI 或函数，并从快捷菜单中选择 **Examples** 打开帮助主题（其中包括该 VI 或函数的范例链接）。单击选板左上角的图钉按钮可锁定该选板。

即时帮助窗口

在 **Context Help** 窗口中，**Click here for more help** 链接和 **More Help** 按钮均已重名为 **Detailed help**。

NI Example Finder 改进

通过 NI Example Finder 的 **Browse** 和 **Search** 栏的 **Requirements** 列表可以显示运行范例所需的其它软硬件。在 **Configure Example Finder** 对话框中，勾选 **General** 栏的 **Show Requirements list** 复选框，以显示 NI Example Finder 的 **Requirements** 列表。

NI Example Finder 可以显示或提交 NI Developer Zone 网上范例的评论和评分信息。NI Example Finder 也包括网上范例相关文档的链接。在 **Configure Example Finder** 对话框中，勾选 **Web** 栏的 **Show Resources list** 复选框可显示 **Resources** 列表。

使用 NI Example Finder 的 **Browse** 栏可浏览 NI Developer Zone 范例，查询用于特定任务并需要使用软硬件产品的各种范例。**(Windows 和 Linux)** 选择 **Browse according to** 部分的 **Task** 选项，并勾选 **Include ni.com examples** 复选框可以浏览网上范例。您可以通过 **Configure Example Finder** 对话框中 **Web** 栏的 **Maximum number of Web query results** 条目，确定可显示网上范例的最大数量。也可通过 **Configure Example Finder** 中 **Web** 栏的 **Web query time limit (seconds)** 条目，确定 NI Example Finder 搜索网上范例的时间限制。

如果通过任务 (Task) 浏览范例，**Double-click an example to open it** 列表中的 **Toolkits and Modules Not Installed** 文件夹将包括用于工具套件和模块的各种当前未安装的范例。如果在 **Toolkits and Modules Not Installed** 文件夹中选择一个范例，**Requirements** 列表将提供一个包括所需工具套件或模块信息的链接。

使用 NI Example Finder 的 **Search** 栏可通过关键字或说明搜索已安装至本地的各种范例。在 **Search** 按钮边上的下拉菜单中选择 **Keywords** 选项，可以使用关键字搜索。在 **Search** 按钮边上的下拉菜单中选择 **Descriptions** 选项，可以搜索范例说明信息。要根据说明搜索范例，必须

得取消勾选 **Include ni.com examples** 复选框。根据说明搜索范例时，在 **Information** 文本框中所搜索的文字将高亮显示。

NI Example Finder 可以搜索或浏览 LabVIEW 中基于 VI 或基于项目的范例。打开基于项目的范例时，LabVIEW 项目文件 (.lvproj) 将在**项目浏览器 (Project Explorer)** 窗口打开。在 **Project Explorer** 窗口中，双击 **My Computer** 中的 VI 即可打开范例。而基于 VI 的范例不会有改动。

在 LabVIEW 7.x 和前期版本中，NI Example Finder 仅仅可以查找已安装的 DAQ 板卡。在 LabVIEW 8.0 中，NI Example Finder 可查找大多数已安装的硬件。单击 **Setup** 按钮显示 **Configure Example Finder** 对话框，也可以将未安装的硬件添加到搜索范围中。

LabVIEW 8.0 其它功能和更改

LabVIEW 8.0 包括以下各种功能和改动。

现有 VI 和函数更改

LabVIEW 8.0 包含以下对现有 VI 和函数的改动：

- VISA Configure Serial Port 和 VISA Serial Break VI 的 **duplicate VISA resource name** 输出端现已更名为 **VISA resource name out**。VISA Get USB Interrupt Data VI 的 **dup VISA USB Intr Event** 输出端现已更名为 **VISA USB Intr Event out**。所有其它 VISA VI 的 **dup VISA resource name** 输出端现已更名为 **VISA resource name out**。
- VISA Assert Trigger 函数的 **protocol** 输入端现已包括两个新值：6 (PXI:Reserve) 和 7 (PXI:Unreserve)。
- VISA Lock Async VI 包括两个新输入端 **lock type** 和 **requested key**，以及一个新输出端 **access key**。通过上述参数可为当前句柄 (Session) 提交专用或共享锁。
- VISA Serial Break VI 的 **Duration** 输入端可以指定毫秒间隔的长度。
- VISA Wait on Event 函数的 **event resource class** 输入端现已更名为 **event class**，并且其属性由必需连接改为可选连接。
- 默认情况下，Get Waveform Components 函数将获取 **Y** 数据数组。
- 如从 **Numeric** 选板中选择 Compound Arithmetic 函数，其默认模式为 “Add”。如从 **Boolean** 选板中选择 Compound Arithmetic 函数，其默认模式为 “OR”。
- To Variant、Variant To Data、Variant To Flattened String 和 Set Variant Attribute 函数的性能大大提高。大多数 VI 可更快地调用上述函数，对于包括大规模数据集的 VI 来说，其性能改善尤其明显。
- Search and Replace String 函数的 **string** 参数现已更名为 **input string**。Search and Replace String 函数还包括 5 个新参数。**ignore case?** 参数用于确定字符串搜索是否区分大小写。**multiline?** 参数用于确定是否将 “^” 和 “\$” 符号分别匹配为行或整个字符串的

开始和结束。 **number of replacements** 参数返回 LabVIEW 替换 **search string** 的次数。该函数同时包括 **error in** 和 **error out** 参数。

- Read PNG File VI 的 **Transparency Thresh** 输入端可将用于 32 位 PNG 图像的 alpha 信息合并至 **image data** 的 **mask** 参数中。LabVIEW 对于任意等于或大于 **Transparency Thresh** 的 alpha 值来说都是不透明的。LabVIEW 对于所有其它 alpha 值完全透明。**Transparency Thresh** 的值必须为 0 至 255。
- Search for Digital Pattern VI 的 **compare mode** 输入端指定搜索时如何处理 X 状态。该 VI 的 DWDT Search for Digital Pattern 实例的 **mode** 和 **start index/seconds** 输入端现已分别更名为 **start value format** 和 **start**。**Index** 模式已更名为 **Samples**。
- Get Y Value VI 中波形实例的 **mode** 和 **index/seconds** 输入端现已分别更名为 **Y position format** 和 **Y position**。该实例的 **actual index/time value** 输出端已更名为 **actual Y position**。
- Get Y Value VI 中 DWDT Get Y Value 实例的 **waveform data value** 输出端可返回波形数据的值。
- Get Y Value VI 中 DTbl Get Y Value 实例的 **digital data value** 输出端可返回数字数据的值。该实例的 **Digital In** 输入端现已更名为 **digital data in**。该实例的 **Digital Out** 和 **Digital Value** 输出端已分别更名为 **digital data out** 和 **digital value**。
- Get Waveform Subset VI 中 WDT Get Waveform Subset 和 DWDT Get Waveform Subset 实例的 **Mode**、**Waveform in**、**Start** 和 **Duration** 输入端现已分别更名为 **start/duration format**、**waveform in**、**start** 和 **duration**。**Index** 模式已更名为 **Samples**。WDT Get Waveform Subset 实例还包括新参数 **start/duration format** 和 **Absolute Time**。Get Waveform Subset VI 中 WDT Get Waveform Subset 和 DWDT Get Waveform Subset 实例的 **Waveform out**、**Actual Start** 和 **Actual Duration** 输出端已分别更名为 **waveform out**、**actual start** 和 **actual duration**。
- Get Waveform Subset VI 中 DTbl Digital Subset 实例的 **Digital**、**Start** 和 **Number of Samples** 输入端已分别更名为 **digital data**、**start** 和 **number of samples**。该实例的 **Digital Subset** 输出端已更名为 **digital data subset**。
- Get Y Value、Waveform to XY Pairs、Index Waveform Array、Align Waveform Timestamps、Append Waveforms、Copy Waveform dt 和 Scale Delta t VI 现已包括支持 64 位有符号整数的实例。Waveform to XY Pairs VI 也包括用于 64 位无符号整数的实例。
- Boolean Array to Digital VI 现包括两个新输入端，其中 **compress data** 输入端指定是否对数字输出进行压缩，**sample rate** 输入端指定输出的数字波形的频率，该频率以采样 / 每秒为单位 (samples per second)。
- Invert Digital VI 的 **response on error** 输入端已更名为 **response to invert error**。Invert Digital VI 还包括两个新输入端。**signal index**

可确定开始转换数据的信号。**number of signals** 指定从信号起始是 **signal index** 开始算起, 需进行转换的信号数量。

- Beep VI 的 **use system alert?** 输入端可确定 LabVIEW 是否使用默认系统警告并忽略 **frequency (Hz)** 和 **duration (msec)** 参数。该 VI 不再包括在 LabVIEW 7.x 和前期版本中使用的 **intensity** 输入端或 **error** 输出端。
- 如果 Save Report to File VI 的 **Prompt to Replace?** 输入端为真 (TRUE) 并且 **report file path** 是已有文件的正确路径, LabVIEW 将打开对话框确认您是否需要替换现有文件。否则 LabVIEW 将在不提示的情况下覆盖现有文件。
- Append Table to Report VI 的默认 **measurement system for column width** 参数现为 **Default**。如使用该 VI 添加表格至 HTML 报告中, LabVIEW 会忽略 **measurement system for column width** 输入端。取而代之的是该 VI 会将 **column width** 输入乘以 100, 以便决定最大列宽度 (像素)。
- Append VI List of SubVIs to Report VI 的 **Include Express VI Configuration Information** 输入端可确定报告中是否包括程序框图上任意 **Express VI** 的配置信息。
- 空间常数 (space constant) 已添加至 **Strings** 选板, 该常数可在程序框图中提供一字节大小的字符。
- Flatten To String 函数包括两个新输入端: **prepend array or string size?** 表示当 **anything** 是一个数组或字符串时, LabVIEW 是否在 **数据字符串 (data string)** 的开始包括数据大小信息, 而 **byte order** 可设置得到的转换字符串 (flattened string) 的 endian 顺序。如果使用上述新输入的默认值, 该函数的使用方式与以前相同。该函数还包括一个新的 **error in** 输入端和新的 **error out** 输出端。
- Flatten To String 函数包括一个新的快捷菜单选项 **Convert 7.x Data**, 该选项可显示 **type string (7.x only)** 输出并在该函数上标注一个红色的 **7.x** 符号。 **Expose Typedefs** 快捷菜单选项只有连接 **type string (7.x only)** 时才可以显示。 **type string** 输出端已更名为 **type string (7.x only)**, 并且只有当已经在 LabVIEW 前期版本中连接该接线端或者右键单击该函数并从快捷菜单中选择 **Convert 7.x Data** 时, 才可以显示该输出端。
- Unflatten From String 函数包括两个新输入端: **data includes array or string size?** 表示 LabVIEW 是否从输入数组或字符串的开始读取数据大小信息, 而 **byte order** 表示输入转换字符串的 endian 顺序。如果使用上述新输入的默认值, 该函数的使用方式与以前相同。该函数也包括一个新输出端 **rest of the binary string**, 其中包括该函数没有进行转换的剩余字节。 **err** 输出端已改为一个新 **error in** 输入端和新 **error out** 输出端。
- Date/Time to Seconds 函数包括一个新输入端 **is UTC**, 用于指定 **date time rec** 是通用时间 (Universal Time) 还是计算机的配置时区。在 LabVIEW 7.x 中, 由于 Date/Time to Seconds 函数根据通用时间进行计算, 因此该函数将忽略 **is DST** 参数。如果 **is UTC** 为 FALSE

(默认), LabVIEW 8.0 会根据 **Is DST** 进行时间转换。打开由 LabVIEW 7.x 或前期版本所保存的 VI 时, LabVIEW 将 **Is DST** 赋值为 -1 以避免转换时间。

- Date/Time to Seconds 和 Seconds to Date/Time 函数在 **date time rec** 簇中包括一个新元素 **fractional second**, 表示秒的小数部分。
- Seconds to Date/Time 函数的 **to UTC** 输入端可确定 **date time rec** 是通用时间还是计算机的配置时区。
- Format Date/Time String 函数包括一个新输入端 **UTC format**, 用于确定输出字符串是通用时间还是计算机的配置时区。
- Format Into String 和 Scan From String 函数可支持时间标识 (time stamp) 数据。 **Edit Format String** 对话框包括 **Format relative time** 和 **Format time stamp** 转换选项, **Edit Scan String** 对话框包括 **Scan relative time** 和 **Scan time stamp** 转换选项。
- \$ 指定格式的语法元素 (format specifier syntax element) 以及 Format Into String 函数可用于确定格式化字符串中的变量顺序。
- Set Property VI 中 String、DBL、SGL、I32、I16、U8 和 Time stamp 实例的 **Create if not found** 输入端用于当所指定的属性在 **Property name** 中并不存在时, 确定是否创建一个自定义属性。

新 VI 范例

关于查看或运行 LabVIEW 8.0 新增的 VI 范例见 NI Example Finder 中 **Browse** 栏的 **New Examples for LabVIEW 8.0** 文件夹。

External Code 函数更改

您可从 DLL 或 CIN 调用下列新函数。

- CToLStr 可将 C 字符串 (string) 转换至 LabVIEW 字符串。
- LToCStr 可将 LabVIEW 字符串 (string) 转换至 C 字符串。
- DSSetHandleFromPtrNULLMeansEmpty 可从指针分配句柄并且可将指针的数据复制到句柄中。该函数与 DSSetHandleFromPtr 函数相似, 只在指针中无数据时两者有所区别, DSSetHandleFromPtrNULLMeansEmpty 函数所生成句柄为 NULL 而不是给该句柄分配 0 字节。
- NIGetOneErrorCode 可将数字错误代码转换为相应的文本描述。

CIN 不再支持在 LabVIEW 7.x 和前期版本中使用的 AZHPurge、AZHNoPurge、AZHLock 或 AZHUnlock 函数。

重命名的属性和方法

LabVIEW 8.0 包括新 VI Server 类、属性、方法和事件。关于上述新 VI Server 项目的详细列表信息见 *LabVIEW Help*。在 LabVIEW 8.0 中已重命名以下属性和方法。

属性

已重命名以下属性。

表 6 LabVIEW 8.0 中重命名的属性

类	LabVIEW 7.1 名称	LabVIEW 8.0 名称
Application	Application:Real-Time Host Connected	Application:User Interface Available
DigitalTable	Cell BG Color	Active Cell:Cell Background Color
DigitalTable	Cell FG Color	Active Cell:Cell Foreground Color
DigitalTable	Top Left Visible Cell	Top Left Cell
Listbox	Scrollbar Visible	Visible Items:Vertical Scrollbar Visible
Listbox	Symbols Visible	Visible Items:Symbols Visible
MulticolumnListbox	Cell Background Color	Active Cell:Cell Background Color
MulticolumnListbox	Cell Size	Active Cell:Cell Size
MulticolumnListbox	Column Headers Visible	Visible Items:Column Headers Visible
MulticolumnListbox	Horizontal Scrollbar Visible	Visible Items:Horizontal Scrollbar Visible
MulticolumnListbox	Symbols Visible	Visible Items:Symbols Visible
MulticolumnListbox	Vertical Scrollbar Visible	Visible Items:Vertical Scrollbar Visible
Scale	Minor Tick Color	Tick Colors:Minor Tick Color
String	Scrollbar Visible	Vertical Scrollbar Visible
Table	Cell BG Color	Active Cell:Cell Background Color
Table	Cell Size	Active Cell:Cell Size
Table	Column Headers Visible	Visible Items:Column Headers Visible
Table	Horizontal Scrollbar Visible	Visible Items:Horizontal Scrollbar Visible
Table	Index Visible	Visible Items:Index Visible

表 6 LabVIEW 8.0 中重命名的属性（续）

类	LabVIEW 7.1 名称	LabVIEW 8.0 名称
Table	Row Headers Visible	Visible Items:Row Headers Visible
Table	Vertical Scrollbar Visible	Visible Items:Vertical Scrollbar Visible
TreeControl	Allow Dragging	Drag/Drop:Allow Item Dragging
TreeControl	Allow Dragging Between Items	Drag/Drop:Allow Dropping Between Items
TreeControl	Allow Dragging of Parent Items	Drag/Drop:Allow Dragging of Parent Items
WaveformChart	Autosize Legend	Legend:Autosize
WaveformGraph	Autosize Legend	Legend:Autosize

方法

已重命名以下方法。

表 7 LabVIEW 8.0 中重命名的方法

类	LabVIEW 7.1 名称	LabVIEW 8.0 名称
Control	Reinit To Dflt	Reinitialize To Default
Listbox	Get DbIClk Row	Get Double-Clicked Row
MulticolumnListbox	Get DbIClk Row	Get Double-Clicked Row
Text	MoveToDefLoc	Move to Default Location
TreeControl	Custom Item Symbols:Revert Symbol	Custom Item Symbols:Revert To Built In Symbol
TreeControl	Custom Item Symbols:Revert Symbols	Custom Item Symbols:Revert All To Built In Symbol
TreeControl	Custom Item Symbols:Set Symbol	Custom Item Symbols:Set To Custom Symbol
TreeControl	Custom Item Symbols:Set Symbol Array	Custom Item Symbols:Set To Custom Symbol Array
VI	Close FP	Front Panel:Close
VI	Export VI Strings	VI Strings:Export
VI	Get All Control Values	Control Value:Get All (Flattened)
VI	Get All Control Values (Variant)	Control Value:Get All (Variant)

表 7 LabVIEW 8.0 中重命名的方法（续）

类	LabVIEW 7.1 名称	LabVIEW 8.0 名称
VI	Get Control Value	Control Value:Get (Flattened)
VI	Get Control Value (Variant)	Control Value:Get (Variant)
VI	Get Diagram Image Scaled	Block Diagram:Get Image Scaled
VI	Get Lock State	Lock State:Get
VI	Get Panel Image	Front Panel:Get Panel Image
VI	Get Panel Image Scaled	Front Panel:Get Panel Image Scaled
VI	Get VI Icon as Image Data	VI Icon:Get As Image Data
VI	Import VI Strings	VI Strings:Import
VI	Lock Remote Panel Control	Remote Panel:Lock Control
VI	Make Current Values Default	Default Values:Make Current Default
VI	Open FP	Front Panel:Open
VI	Reinitialize All To Default	Default Values:Reinitialize All To Default
VI	Save VI Icon to File	VI Icon:Save To File
VI	Set Control Value	Control Value:Set (Flattened)
VI	Set Control Value (Variant)	Control Value:Set (Variant)
VI	Set Lock State	Lock State:Set
VI	Set VI Icon from File	VI Icon:Set From File
VI	Set VI Icon from Image Data	VI Icon:Set From Image Data
VI	Unlock Remote Panel Control	Remote Panel:Unlock Control

错误代码改进

现在可以使用错误代码 -8999 到 -8000 说明自定义错误信息。

LabVIEW 8.0 包括以下错误代码 (error code) 更改。

表 8 LabVIEW 8.0 错误代码更改

LabVIEW 7.1 错误代码	LabVIEW 8.0 错误代码
20003	20012
20101	20111
20102	20112
20103	20113
20104	20114

其它

LabVIEW 8.0 中还包括以下改动：

- 在 LabVIEW 前期版本中，运行未注册的 LabVIEW 时总会出现 **Registration Information** 对话框，而在 LabVIEW 8.0 中不会出现该对话框。LabVIEW 8.0 将在安装过程中收集个人信息，如您的姓名和公司名称等。
- (Windows)**在LabVIEW 7.x和前期版本中，按住<Ctrl-Tab>键可根据打开 LabVIEW 窗口的顺序在各个窗口间进行循环切换。在 LabVIEW 8.0 中，按住 <Ctrl-Tab> 键将根据 LabVIEW 窗口出现在屏幕上的顺序进行各个窗口间的循环切换。该顺序与按 <Alt-Tab> 键所看到的顺序相同。**(Linux)** 窗口的顺序取决于所使用的窗口管理器 (window manager)。
- 如将一个条目拖放至树形控件的 child-only 条目下，LabVIEW 会将该条目置于 child-only 条目下方相同层次等级的位置。
- 使用 VI Refnum(面板图像)探针可显示 VI 名称、路径、VI 前面板图像和 VI 引用的十六进制值。如该值为非法的引用句柄，可以使用该探针设置一个断点。
- 如所设置属性的项目无法在列表框中找到，禁用列表框和多列列表框属性时将返回错误信息。
- (Mac OS)** LabVIEW 可支持鼠标滚轮操作。查看条件、事件和层叠式顺序结构 (Case, Event, and Stacked Sequence structure) 的子程序框图时，通过将光标移至选择器标签，并在移动鼠标滚轮的同时按住 <命令> (Command) 键，就可以滚动浏览上述子程序框图。
- (Mac OS)** LabVIEW 仅支持弹出窗口的 <Control> 单击，而不再支持 <Command> 单击按钮组合。
- 在 **Control Editor** 窗口中使用自定义模式时，可使用快捷菜单的 **Import from File** 选项从文件中导入图片作为自定义控件的一部分。通

过选择修饰 (decoration) 并单击 **Edit»Paste**, 可以使用剪贴板中的图片替换修饰。右键单击该装饰并从快捷菜单中选择 **Import from File** 可使用文件中的图片替换该修饰。

- 在 **Options** 对话框的 **Paths Options** 选项卡中无法指定一个自定义的**菜单目录 (Menus Directory)**。
- 在 **Paths Options** 选项卡中指定自定义的**VI 搜索路径 (VI Search Path)** 时, 现可使用 `<osdatadir>` 符号路径。
- 将时间标记转换为变量 (variant) 时, 该变量显示控件会显示所连接的时间标记的当前值。
- LabVIEW 8.0 不再提供 CVI Function Panel Converter 和 **Update VXIPNP Drivers** 对话框。从 ni.com/idnet 下载用于 LabWindows™/CVI™ 仪器驱动程序工具的 LabVIEW 界面生成器 (LabVIEW Interface Generator for LabWindows/CVI Instrument Drivers tool) 以便实现该功能。
- **End text entry with Enter Key** 选项现已包括在 **Options** 对话框的 **Environment** 选项卡中。
- **Explain Changes** 对话框现已包括 **Path** 文本框, 用于列出所选 VI 的磁盘路径。使用 **Path** 文本框可区分名称相同的 VI。
- 对 VI 的调试禁用功能已有所改进。
- **VI Library Manager** 窗口现更名为 **LLB Manager** 窗口。右键单击 **Files** 列表中任一条目, 并从快捷菜单中选择 **Top Level?** 可确定所选 LLB 中的条目是否位于最上层。无法取消在 **LLB Manager** 窗口中所作的改动。
- **Edit VI Library** 对话框现已从 **Tools** 菜单中删除。通过 **LLB Manager** 窗口, 可删除库中的 VI 或者将库中的 VI 标为最上层 VI。
- 在 LabVIEW 7.x 和前期版本中, 如将一个错误簇连接到条件结构, True 条件分支将转变为 Error 条件分支, 而 False 条件分支转变为 No Error 条件分支。在 LabVIEW 8.0 中, 如将一个错误串连接到条件结构, True 条件分支转变为 No Error 条件分支, 而 False 条件分支转变为 Error 条件分支。
- 要在程序框图上创建特定的属性或方法, 可以右键单击前面板的输入控件或显示控件, 选择 **Create»Property Node** 或 **Create»Invoke Node**, 并从快捷菜单中选择所需属性或方法。
- **(Windows)** 使用 **CAN Controls** 选板中的 CAN Channel 控件以及 NI-CAN, 就可以访问 NI-CAN Channel API。关于 **CAN Controls** 选板中控件的详细信息见 NI-CAN 文档中的 *CAN Initialize* 主题。
- 在 **Properties** 对话框的 **Format and Precision** 选项卡上, 当 **Precision Type** 设置为 **Significant digits** 时, National Instruments 建议您在用于单精度时在 **Digits** 栏中取值范围为 1 到 6, 用于双精度、扩展精度和浮点数时使用 1 到 13。
- 在图表 (Chart)、强度图 (Intensity Chart)、强度图形 (Intensity Graph) 的 **Properties** 对话框中 **Scales** 选项卡上, 勾选 **Ignore waveform stamp on x-axes** 复选框, 即可将 X 轴作为横坐标轴并将

其初始值设置为 0 而不是特定 t0 值。选择 Y 轴作为纵坐标轴并将数字波形数据作为数据总线的各个数据线显示时，只需勾选 **Expand digital buses** 复选框。

- 可以一次打开多个文件。(Windows) 选择 **File»Open** 可显示标准文件对话框，并按住 <Shift> 键或 <Control> 键选择多个文件。(Mac OS) 使用 <Shift> 或 <Option> 键。(Linux) 使用 <Shift> 或 <Alt> 键。
- 所有浏览器可支持 **Web Publishing Tool** 对话框的 **Monitor** 选项。
- 在 LabVIEW 7.x 和前期版本中，选择 **Tools»Compare»Compare VIs** 并单击 **Select** 按钮时，将出现 **Select VI By Name** 对话框。在 LabVIEW 8.0 中，选择 **Tools»Compare»Compare VIs** 并单击 **Select** 按钮时，将出现 **Select a VI** 对话框。
- 在 LabVIEW 8.0 中，选择 **Tools»Compare»Compare VI Hierarchies** 时，可以从文件系统或内存中的 VI 列表中选择所需 VI。
- 在 LabVIEW 7.x 中，您可以使用右键单击任何程序框图的对象并从快捷菜单中选择 **Source Palette**，可查看子选板中的类似对象，其中 **Source** 为包含该程序框图对象的子选板名称。在 LabVIEW 8.0 中，也可以右键单击程序框图对象并从快捷菜单中选择 **Replace»Source Palette**，从而使用子选板的类似对象替换该程序框图对象，其中该子选板已包含所需程序框图对象。也可以右键单击连接线并从快捷菜单中选择 **Insert»Source Palette**，插入源选板中最常使用的对象。该选项将在连接线所连接的对象之间插入所选对象。
- 在 LabVIEW 7.x 和前期版本中，当修改自定义类型的数据类型以及更新自定义类型的实例时，该实例将仅保留当前数据、默认数据、专用数据、标签和说明。在 LabVIEW 8.0 中，自定义类型的实例可保留更多属性。
- 在 LabVIEW 8.0 中，将自定义类型换成另一控件时，相对于 LabVIEW 7.x 和前期版本，该控件所保留的自定义类型属性可能更少。
- LabVIEW 推荐使用自定义类型。
- **Set Width and Height** 对话框现已更名为 **Resize Objects** 对话框。在该对话框的 **Current Size Values** 列表框中，对象值后面的星号大小不会由于 **Width** 或 **Height** 调整而改变。
- 在 LabVIEW 7.x 中，如果 LabVIEW 无法找到 VI，必须使用相同名字的 VI 替换缺少的 VI。在 LabVIEW 8.0 中，可以使用任意名字的 VI 替换所缺少的 VI。LabVIEW 将使用您所选 VI 替换缺少 VI 的所有实例。
- 在 LabVIEW 7.x 和前期版本中，通过按 <Ctrl-A> 键可以对前面板或程序框图中的对象重复进行对齐操作。在 LabVIEW 8.0 中，按 <Ctrl-Shift-A> 键时可以重复对齐操作。而按 <Ctrl-A> 键时可在前面板或程序框图中选择所有对象。
- LabVIEW 可以配置加载子 VI 的时间。右键单击某一个子 VI 并从快捷菜单中选择 **Call Setup**，显示 **VI Call Configuration** 对话框。该快捷菜单选项仅适用于由可支持 VI Server 调用的程序实例所打开的 VI，而无法在函数、Express VI、多态 VI 以及由不支持 VI Server 调用的程序实例所打开的 VI 中使用，如 PDA 和 FPGA 终端。如果有一个大的主 VI，通过在对话框中选择 **Load and retain on first call** 可节省加载时

间和内存。选择该选项时，子 VI 仅在主 VI 调用时才进行加载，并在操作结束后可从内存中释放该子 VI。

- 在 LabVIEW 8.0 中，从程序框图中复制前面板控件并将其粘贴到新 VI 的程序框图中时，前面板中的控件布局将和原 VI 不同。该对象将会置于前面板的左上角，从而避免对象重叠或将对象复制到不可见区域中。
- LabVIEW 可显示年份的时间范围已从 1904 至 2038 扩展为 1600 至 3000。
- 如需在 **Context Help** 窗口中显示一个回车，必须使用两个回车进行分段。
- 如果按键与 VI 菜单中的键盘快捷键相匹配（如 <Ctrl-C> 或 <Ctrl-V>），无论该菜单选项是否启用，LabVIEW 都不会产生 Key Down 事件。
- Application:Language 属性可返回 **ko**，用于表示 LabVIEW 开发环境的语言为韩语。
- VIs Strings:Export 和 VIs Strings:Import 方法可分别用于将多个 VI 导出到标记文本文件或者从标记文本文件中导入多个 VI。

National Instruments、NI、ni.com 和 LabVIEW 为 National Instruments Corporation 的商标。有关 National Instruments 商标的详细信息见 ni.com/legal 上的 *Terms of Use* 部分。此处提及的其它产品和公司名称为其各自公司的商标或商业名称。关于 National Instruments 产品的专利权，见软件中 **Help»Patents**，CD 中 `patents.txt` 文档，或登录 ni.com/patents。USI (Xerces C++、ICU 和 HDF5) 中使用组件的版权、使用条件和免责条款见 `USICopyrights.chm`。