

LabVIEW™ 2017升级说明

本文档介绍在Windows、OSX和Linux操作系统上升级至LabVIEW 2017的步骤。升级之前，请阅读本文档以了解下列内容：

- LabVIEW升级的推荐流程
- 加载较早版本LabVIEW中保存的VI可能遇到的兼容性问题
- LabVIEW 2017的新增功能和改动

目录

升级至LabVIEW 2017.....	1
1. 备份VI和机器配置.....	2
2. 测试和记录VI的当前操作.....	3
3. 安装LabVIEW、附加软件包和设备驱动.....	3
4. 转换VI并修正VI操作上的改动.....	3
常见升级问题疑难解答.....	5
升级和兼容性问题.....	5
从LabVIEW 2012或更早版本升级.....	5
从LabVIEW 2013升级.....	5
从LabVIEW 2014升级.....	6
从LabVIEW 2015升级.....	7
从LabVIEW 2016升级.....	7
LabVIEW 2017的新增功能及改动.....	7
减少VI加载及编译时间.....	7
移动对象时保持连线连接.....	7
自适应VI.....	7
新增和改动的VI和函数.....	8
新增和改动的类、属性、方法和事件.....	8
应用程序生成器的改进.....	8
前期版本LabVIEW的改动.....	9

升级至LabVIEW 2017

安装较新版本的LabVIEW然后用新版LabVIEW打开已有的VI并保存，可以将较早版本的VI升级到新版本。NI建议用户使用更有效率的升级流程，快速检查和纠正升级过程中出现的错误。



提示 升级流程对具备下列特征的LabVIEW大型应用程序至关重要：控制或监控关键操作、停机代价高、使用多个模块、工作包和驱动、程序版本为不支持的LabVIEW版本，建议使用推荐流程进行升级。关于各个版本LabVIEW的支持情况，请访问NI网站 ni.com/info 并输入信息代码 `lifecycle` 查询。

推荐升级流程的概述

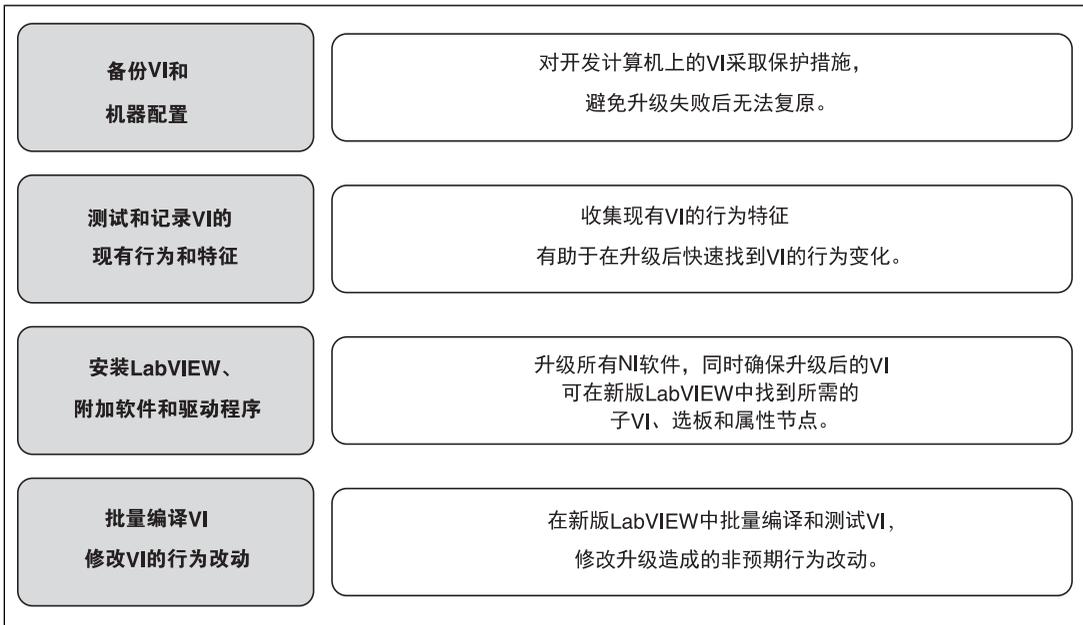


图 1



注： 如从LabVIEW 5.1或更早版本升级，必须先升级到某个中间版本。关于从特定版本LabVIEW升级的详细信息，请访问ni.com/info，输入信息代码upgradeOld查询。

1. 备份VI和机器配置

升级至LabVIEW 2017之前先备份VI和机器配置，可保证在必要的时候恢复VI为前期版本或重新开始升级过程。

a. 备份VI

升级之前备份VI可在意外情况下将VI快速恢复为升级前的版本。升级完成后，无法在前期版本的LabVIEW中打开升级的VI。需在LabVIEW中将VI保存为前期版本，才能使用前期版本的LabVIEW打开VI。

可通过下列方法备份VI：

- **提交VI至源代码控制软件**—如更新VI后，VI的行为发生了不可预期的改变，可恢复为源代码控制软件中保存的版本。关于在LabVIEW中使用源代码控制软件的详细信息，见LabVIEW帮助目录栏的**基础»使用项目和终端»概念»在LabVIEW中使用源代码控制**。
- **创建VI的副本**—通过不同方法复制VI：
 - 保存为项目—打开项目并选择**文件»另存为**，复制.lvproj文件和所有项目内容。选择**包括全部依赖关系**，确保也保存了项目的依赖文件。
 - 另存为LLB或目录中的VI—从操作系统的文件浏览器中创建LLB或目录的备份，并将备份保存在与原文件不同的位置。为了避免名称冲突，请不要将副本保存在同一个磁盘上。

b. 备份计算机配置

安装新版LabVIEW会影响共享文件，从而改变前期版本VI的行为。但是，更新共享文件后，很难将更新后的共享文件恢复为前期版本。因此，可考虑下列使用方法备份开发计算机上的NI软件配置：

- **创建机器配置的备份镜像**—使用磁盘镜像软件保存升级前磁盘的状态，包括已安装的软件、用户设置和文件。如要在升级后恢复磁盘镜像，应用备份镜像即可。
- **在测试计算机上测试升级流程**—在测试机上的升级测试比创建备份镜像所需的时间更长，但是仍然建议为控制或监测生产的设备采取此步骤。解决计算机升级测试的问题后，可将测试机作为生产主机，或在生产主机上重复升级流程。



提示 要尽量减少VI在开发计算机和测试机之间运行的差别，请使用硬件配置软件环境尽量和开发计算机接近的测试机。

2. 测试和记录VI的当前操作

更新VI时，LabVIEW版本之间的改动可能会影响VI的操作。在升级前后两个版本中测试，可比较升级对VI操作产生的影响。请确保得到下列测试的结果：

- **批量编译记录**—在前期版本中批量编译VI可生成断开VI的详细记录。如VI由多人开发和维护，或者部分VI最近没有编译，升级前请批量编译VI。勾选**批量编译**对话框的**记录结果**复选框，即可生成批量编译记录。关于批量编译VI的详细信息，请参考LabVIEW帮助中目录栏的**基础»创建VI和子VI»详解»保存VI»批量编译VI**。
- 验证各个VI是否实现了其各自功能的单元测试
- 验证VI整体运行情况的集成测试
- 验证VI部署到终端后运行情况的部署测试
- 性能测试，包括CPU使用、内存使用和执行速度等。可使用**性能和内存信息**窗口获取VI性能的数据。
- 检测VI是否能处理预期外的数据的压力测试

关于测试VI的详细信息，请参考LabVIEW帮助目录栏的**基础»应用程序开发和设计规范»概念»开发大型应用程序»开发模型的各个阶段»测试应用程序**。



注： 测试中如修改了任何VI，进行下一步之前请先备份VI。

3. 安装LabVIEW、附加软件包和设备驱动

a. 安装LabVIEW、模块、工具包和驱动程序

升级到新版LabVIEW时，不仅需要升级新的LabVIEW开发系统，还要将模块、工具包和驱动程序升级到与新版LabVIEW兼容的版本。

b. 复制user.lib文件

要使用在上一个版本LabVIEW中创建的自定义控件和VI，可将上一个版本LabVIEW目录下的labview\user.lib子目录复制到LabVIEW 2017安装目录下的labview\user.lib子目录。

4. 转换VI并修正VI操作上的改动

在LabVIEW 2017中批量编译VI会将VI保存为最新的版本，并创建一个错误记录用于找到断线的VI。参考记录信息与升级和兼容性问题部分，检查并修改版本迁移过程中出现的问题。

a. 在新版LabVIEW中批量编译VI

批量编译VI的同时，VI被保存为LabVIEW 2017版本。批量编译后，之前版本的LabVIEW无法打开这些VI。需在当前LabVIEW中打开该VI或项目，选择**文件»保存为前期版本**，才能继续使用之。

前版本的LabVIEW打开这些VI。因此，请只批量编译需要转换为新版本LabVIEW的VI。为便于找出升级引发的问题，可勾选**批量编译**对话框中的**记录结果**复选框，创建一个批量编译记录。



注： 批量编译包含FPGA或实时资源的VI时，**批量编译**对话框可能将VI报告为不可执行的VI。如需检查错误，必须在LabVIEW项目（包含所需FPGA或实时资源）的FPGA或RT终端下打开VI。

关于批量编译VI的详细信息，请参考LabVIEW帮助目录栏的下列主题：

- **基础»创建VI和子VI»详解»保存VI»批量编译VI**
- **基础»创建VI和子VI»详解»保存VI»批量编译的常见状态信息**

b. 修复断线VI

如使用了修改过的功能，前期版本LabVIEW和LabVIEW 2017之间的差别可能会导致某些VI出现断线。按照下列步骤快速找到并解决LabVIEW 2017 VI中的断线：

1. 将批量编译记录与测试现有VI的记录比较，找到升级过程中有断线的VI。
2. 要判断是否因LabVIEW升级造成VI断开，请参考本文档的升级和兼容性问题。

c. 找到并修改行为改动

NI努力将各个版本LabVIEW之间的行为差别减至最小，但是版本之间的改进还是会影响VI的行为。可使用下列工具，快速找到VI是否有行为改动：

- **升级VI分析器测试**—该测试用于快速找到大量VI升级后是否有行为改动。按照下列步骤获取和使用下列测试：
 1. 下载升级分析器测试，请访问NI网站ni.com/info并输入信息代码analyzevi。
 2. 选择**工具»VI分析器»分析VI**，开始一个VI分析任务。如从**项目浏览器**窗口而不是VI窗口选择上述菜单选项，可一次分析整个项目。
 3. 请参考相关版本的升级和兼容性问题部分，解决失败的测试。例如，如LabVIEW 2014升级VI分析器测试找到了一个行为改动，请参考从LabVIEW 2013升级。
- **升级文档**
 - 升级和兼容性问题—该部分列出了版本升级对VI行为的影响。请在升级前仔细阅读相关版本的LabVIEW升级和兼容性问题。



提示 要快速找到升级和兼容性问题中提到的对象，请打开升级后的VI并选择**编辑»查找和替换**。

- **LabVIEW 2017已知问题**—该部分列出了LabVIEW 2017以及之前版本中所有已知程序漏洞。访问NI网站ni.com/info并输入信息代码lv2017ki，可查看该列表。请参考升级（Upgrade）—行为改动（Behavior Change）和升级（Upgrade）—版本迁移（Migration）（如存在）部分找到可能影响升级VI的程序漏洞。
- **模块和工具包文档**—列出特定模块和工具包（例如，LabVIEW FPGA模块和LabVIEW Real-Time模块）的更新问题。
- **驱动程序自述文档**—各个驱动程序的升级问题。自述文档在驱动程序的安装光盘上。



提示 要判断VI行为变更是因为驱动升级，而不是LabVIEW升级，请于安装LabVIEW 2017后在早期LabVIEW版本中测试这些VI。

- **自定义测试**—在LabVIEW 2017中打开VI，进行与在前期版本中同样的测试，比较结果。如有行为改动，请参考升级说明。

常见升级问题疑难解答

关于解决下列升级问题的详细信息，请参考LabVIEW帮助中目录栏的[升级至LabVIEW 2017»常见升级问题疑难解答](#)。

- 找到模块和工具包的功能
- 查找丢失的子VI、选板和属性节点
- LabVIEW 2017为什么无法打开在前期版本LabVIEW中保存的VI
- 判断安装的NI软件版本
- 恢复VI为前期版本

升级和兼容性问题

请参考各个版本LabVIEW中可能改变VI行为的改动。

关于LabVIEW 2017的现有问题记录、其他兼容性问题和新功能的相关信息，见labview目录中的readme.html文件。

从LabVIEW 2012或更早版本升级

关于从LabVIEW 2012或更早版本升级至LabVIEW 2017可能遇到的升级和兼容性问题，请访问NI网站ni.com/info并输入信息代码upnote13zhs查询。关于升级过程中可能遇到的其他问题，请参考本文档的从LabVIEW x升级部分。

从LabVIEW 2013升级

从LabVIEW 2013升级到LabVIEW 2017时可能会遇到下列兼容性问题。关于可能遇到的其他升级问题，请参考从LabVIEW 2014升级、从LabVIEW 2015升级和从LabVIEW 2016升级。

“字符串至路径转换”函数的改进

在LabVIEW 2014及更高版本中，字符串至路径转换函数在读取字符串变体时，如<Not A Path>的变体，不区分大小写，始终返回<Not A Path>的常量值。用户可在字符串输入端指定<not a path>或<Not A Path>，函数将始终返回<Not A Path>的常量值。关于“字符串至路径转换”函数在LabVIEW早期版本中的行为，请参阅下表。

LabVIEW 2012和2013	LabVIEW 2011和更早版本
无论变体如何，“字符串至路径转换”函数不返回<Not A Path>的常量值。用户可指定<Not A Path>的变体，函数将返回指向名为<Not A Path>目录的路径，但不返回<Not A Path>的常量值。	在LabVIEW 2014及更高版本中，“字符串至路径转换”函数不区分大小写。指定字符串的变体时，如<Not A Path>的变体，函数将始终返回<Not A Path>的常量值。例如，无论指定的是<not a path>或<Not a Path>，函数都将返回<Not A Path>的常量值。

检查并更新自定义类型

从自定义类型检查并更新快捷菜单替换了LabVIEW 2013及更早版本中的从自定义类型更新快捷菜单。

不再支持的VI、函数和节点

LabVIEW 2014及更高版本不再支持下列VI、函数和节点：

Apple Event VI

(OS X) LabVIEW 2014及更高版本不再支持Apple Event VI。请使用库与可执行程序选板上的Run AppleScript Code VI与LabVIEW之外的OS X应用程序进行通信。如尝试运行任何包含下列Apple Event VI的VI，LabVIEW可能产生错误且无法运行该VI：

- AESend Do Script
- AESend Finder Open

- AESend Open
- AESend Open Document
- AESend Print Document
- AESend Quit Application
- Get Target ID
- AESend Abort
- AESend Close
- AESend Open, Run, Close
- AESend Run
- AESend VI Active?
- AECreat Comp Descriptor
- AECreat Descriptor List
- AECreat Logical Descriptor
- AECreat Object Specifier
- AECreat Range Descriptor
- AECreat Record
- AESend
- Make Alias

操作者框架VI

LabVIEW 2014及更高版本不再支持“操作者:启动操作者”VI。请使用操作者:启动根操作者VI或操作者:启动嵌套操作者VI。

“读端口”和“写端口”VI

LabVIEW 2014及更高版本不再支持“读端口”和“写端口”VI。

不再支持的属性、方法和事件

LabVIEW 2014及更高版本不再支持应用程序类的“获取VI:旧帮助信息”方法。请使用获取VI:帮助信息方法从VI属性对话框的说明信息页返回指定VI的帮助信息。

从LabVIEW 2014升级

从LabVIEW 2014升级到LabVIEW 2017时可能会遇到下列兼容性问题。关于可能遇到的其他升级问题，请参考从LabVIEW 2015升级和从LabVIEW 2016升级。

识别LabVIEW应用程序中的缓冲区分配

LabVIEW 2014 SP1及更高版本包含**监测缓冲区分配**窗口，用于识别和分析LabVIEW应用中的缓冲区分配。选择**工具»性能分析»监测缓冲区分配**，可显示该窗口。

自由标签中的超链接

LabVIEW 2015及更高版本可检测自由标签中的URL并将其转换为带下划线蓝色文本的超链接。从LabVIEW 2014及更早版本升级时，LabVIEW不会自动将自由标签中的URL转换为超链接。如需启用前面板标签的超链接，可右键单击自由标签并在快捷菜单中选择**启用超级链接**。无法禁用程序框图标签中的超链接。

不再支持的VI、函数和节点

LabVIEW 2015及更高版本不再支持下列VI、函数和节点：

- 读取电子表格文件—现使用读取带分隔符电子表格VI。
- 写入电子表格文件—现使用写入带分隔符电子表格VI。

从LabVIEW 2015升级

从LabVIEW 2015升级到LabVIEW 2017时可能会遇到下列兼容性问题。关于可能遇到的其他升级问题，请参考从LabVIEW 2016升级。

在LabVIEW 2016及更高版本中，**快速放置配置**对话框包含前面板和程序框图对象快捷方式的默认列表。在LabVIEW 2015或更早版本中创建的快捷方式不会自动迁移至LabVIEW 2016及更高版本的快捷方式列表。

从LabVIEW 2016升级

从LabVIEW 2016升级到LabVIEW 2017时可能会遇到下列兼容性问题。

操作者框架VI中的行为改变

在LabVIEW 2016及更早版本中，当嵌套操作者由于启动前初始化方法发生错误而启动失败时，嵌套操作者将返回错误，并向其调用方操作者发送包含错误的“最近一次确认”消息。在LabVIEW 2017中，嵌套操作者返回错误，但不向其调用方操作者发送“最近一次确认”消息。

LabVIEW 2017的新增功能及改动

观点交流符号表示来自NI观点交流论坛的产品意见和建议。请登录NI网站ni.com/info并输入信息代码ex3gus，访问NI观点交流论坛。

关于LabVIEW 2017的已知问题、部分已修正问题、其他兼容性问题和新增功能的相关信息，请参考labview目录下的readme.html文件。

减少VI加载及编译时间

(Windows) 对于LabVIEW 2017，NI采用了更为激进的编译器，用于构建LabVIEW开发环境和LabVIEW运行引擎。该升级减少了聚合VI的加载时间及VI编译时间。

移动对象时保持连线连接

在程序框图上将对象移入和移出结构时，LabVIEW 2017将自动保持连线连接。如移入或移出结构的对象已连接至结构内的对象，LabVIEW将创建或移除通道以保持连线连接。移动对象时按下<W>键，可切换自动连线连接。

自适应VI

 LabVIEW 2017包含内嵌至其调用方VI的自适应VI (.vim)，该VI可将每个接线端调整为相应的输入数据类型。通过自适应VI，您可创建VI，对任意可接收的数据类型执行相同的操作，而无需为每种数据类型保存单独的VI副本。

自适应VI类似于多态VI，但在确定可接收的数据类型时，自适应VI更加灵活。多态VI使用一系列预定义的可接收数据类型。自适应VI计算实现是否接收某种数据类型。

自适应VI使用.vim文件扩展名。如需创建自适应VI，可选择**文件»新建**，并在**新建**对话框中选择**自适应VI**。如需将现有VI转换为自适应VI，可将文件扩展名设置为.vim并保存。



注： 仅可将标准VI转换为自适应VI。无法将多态VI、全局VI或XControl功能转换为自适应VI。

内置的自适应VI

LabVIEW提供以下自适应VI，供用户在应用程序中使用。内置自适应VI的图标使用橘色背景。

- **数组选板**
 - **数组元素减1**—将一维数组的指定元素减1。如数组为时间标识数组，该VI将元素减1秒。

- **数组元素加1**—将一维数组的指定元素加1。如数组为时间标识数组，该VI将元素加1秒。
- **重排一维数组**—以伪随机顺序重新排列一维数组元素。
- **重排二维数组**—以伪随机顺序重新排列二维数组元素。
- **排序二维数组**—升序排列指定列或行中的元素，重新排列二维数组的行或列。
- 比较选板
 - **值改变**—如首次调用VI，或输入值与上一次调用VI时发生改变，返回TRUE。
- 转换选板
 - **数值至枚举**—查找匹配指定数值的枚举值，并返回对应的枚举项。
- 定时选板
 - **暂停数据流**—将连线的数据流延迟指定的时间长度。

关于自适应VI的信息，见LabVIEW帮助中目录栏的**基础»创建VI和子VI»概念»创建模块化代码»自适应VI**。

关于使用自适应VI的范例，见labview\examples\Malleable VIs\Basics\Malleable VIs Basics.lvproj。

(NI论坛用户DanyAllard提供的建议。)

新增和改动的VI和函数

LabVIEW 2017包含下列新增和改动的VI和函数：

数据值引用的只读访问

元素同址操作结构的数据值引用读取/写入元素边框节点允许对数值引用进行只读访问。右键单击结构右侧的边框节点，选择**允许并行只读访问**。右侧边框节点未连线时，LabVIEW允许多个并行只读操作，且不修改数值引用。

新通道模板

LabVIEW 2017包含事件消息器通道模板。使用该通道，从多个写入方向一个或多个事件结构传输数据。每次对通道的写入操作将触发一个事件。事件消息器通道允许通道语法与事件语法相结合，控制用户界面事件及生成的事件。关于使用事件消息器通道的范例，请参考labview\examples\Channels\Event Messenger\Channel - Event Messenger.lvproj。

新增和改动的类、属性、方法和事件

LabVIEW 2017包含对获取VI依赖关系（名称和路径）方法的改动。**保留Express VIs?** 参数已重命名为**保留Express和自适应VI?**。如**保留Express和自适应VI?**为FALSE（默认），LabVIEW返回位于Express VI和自适应VI之下的隐藏实例VI的名称。如值为TRUE，LabVIEW返回Express VI和自适应VI作为依赖关系。如希望得到编辑时依赖关系，可将**保留Express和自适应VI?**设为TRUE。如希望得到运行时依赖关系，可将**保留Express和自适应VI?**设为FALSE。无论如何设置该项，LabVIEW都将包括实例VI的子VI，作为引用VI的依赖关系。

应用程序生成器的改进

LabVIEW 2017对LabVIEW应用程序生成器和程序生成规范进行了下列改进。

LabVIEW运行时引擎的向后兼容性

在以前的LabVIEW版本中，无法在不重新编译的情况下加载和运行旧版本LabVIEW中构建的二进制文件和VI。从2017开始，LabVIEW支持LabVIEW运行引擎的向后兼容性。例如，LabVIEW 2017以后的版本可加载LabVIEW 2017创建的二进制文件及VI，无需重新编译。该改进适用于独立应用程序(EXE)、共享库(DLL)以及打包项目库。

要使二进制文件向后兼容，请根据您的程序生成规范，勾选特定对话框高级页面上的以下复选框：

程序生成规范	对话框	复选框
独立应用程序 (EXE)	应用程序属性	允许未来版本的LabVIEW运行引擎运行该应用程序
打包项目库	打包库属性	允许未来版本的LabVIEW加载该打包库
共享库 (DLL)	共享库属性	允许未来版本的LabVIEW加载该共享库

对于在LabVIEW 2017及以后版本中创建的程序生成规范，LabVIEW默认启用这些选项。您可禁用这些选项，将程序生成规范绑定至特定版本的LabVIEW。禁用这些选项将防止对性能配置文件进行任何更改，并帮助您避免编译器升级导致的意外问题。对于实时应用程序，对话框中不显示这些选项，但功能默认为启用。

LabVIEW和其他语言之间的调用的改进

LabVIEW 2017中，LabVIEW生成的共享库 (DLL) 的性能和稳定性得到大幅提升，尤其是对于从LabVIEW和其他语言对LabVIEW生成的DLL的调用而言。例如，从C语言应用程序对LabVIEW生成的DLL的调用可在多线程执行系统中运行。这些改进还防止了从LabVIEW调用LabVIEW生成的DLL时可能发生的死锁和原子性违规。

如需使用该功能，在**共享库属性**对话框的高级页中勾选在**私有执行系统中执行VI**复选框。默认为新程序生成规范启用该选项。该选项对于从LabVIEW 2016及更早版本迁移的程序生成规范禁用，以防止行为发生意外更改。例如，当从非LabVIEW应用程序调用LabVIEW生成的共享库时，禁用该选项可阻止依赖单线程执行的共享库在多线程执行系统中执行。(NI Linux Real-Time) 默认情况下，该选项对Linux RT终端禁用，以避免潜在的性能抖动。

前期版本LabVIEW的改动

要查看LabVIEW自您所安装版本以来的完整新增和改动的内容，请参考历次发布的升级说明。访问NI网站ni.com/info，输入下列信息代码：

- LabVIEW 2013升级说明—[upnote13zhs](#)
- LabVIEW 2014升级说明—[upnote14zhs](#)
- LabVIEW 2015升级说明—[upnote15zhs](#)
- LabVIEW 2016升级说明—[upnote16zhs](#)

关于NI商标的详细信息，请访问ni.com/trademarks，查看NI Trademarks and Logo Guidelines页面。此处提及的其他产品和公司名称均为其各自公司的商标或商业名称。关于NI产品和技术的专利权，请查看软件中的帮助»专利信息、光盘中的patents.txt文件，或ni.com/patents上的National Instruments Patent Notice。产品安装结束后，可在自述文件中查看最终用户许可协议 (EULA) 和第三方法律声明。请登录ni.com/legal/export-compliance上的Export Compliance Information查阅NI全球出口管制政策，以及如何获知有关的HTS编码、ECCN和其他进出口信息。NI对于本文件所含信息的准确性不作任何明示或默示的保证，并对其错误不承担任何责任。美国政府用户：本手册中包含的数据系使用私人经费开发的，且本手册所包含的数据受到联邦采购条例52.227-14和联邦国防采购条例补充规定252.227-7014和252.227-7015中规定适用的有限权利和受限数据权益条款的约束。