LabVIEW™ 2018升级说明

本文档介绍在Windows、macOS和Linux操作系统上升级至LabVIEW 2018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 2018的新增功能和改动

目录

升级至LabVIEW 2018

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的出现的错误。

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。

NATIONAL INSTRUMENTS™

推荐升级流程的概述

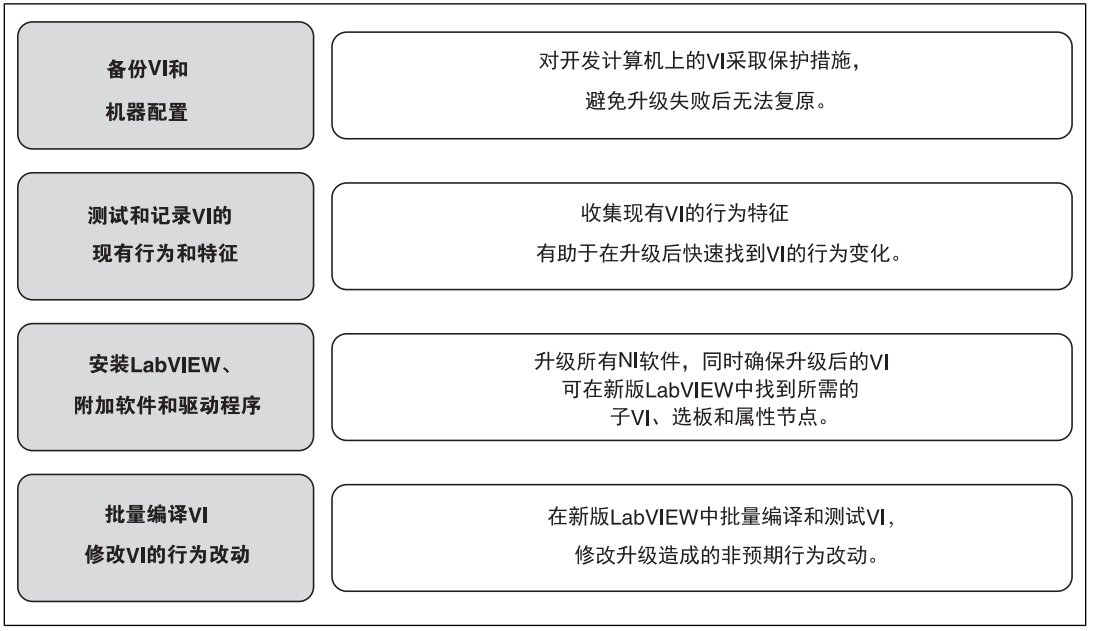

图 1

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 2018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的行为。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 2018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 2018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。

注： 如果在升级到LabVIEW 2017时批量编译了VI，则可以跳过这一步，因为LabVIEW 2017和更高版本支持LabVIEW运行引擎的向后兼容性。

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 2018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 2018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 2018 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：

- **运行VI分析器升级测试**—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. 为比先前版本更新的LabVIEW下载VI分析器升级测试。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 2015 VI分析器升级测试检测到一个行为改动，请参考从LabVIEW 2014升级的相关主题。
- 阅读升级文档
  - 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 2018已知问题－该部分列出了LabVIEW 2018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`lv2018ki`，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程序升级，而不是LabVIEW升级，请于安装LabVIEW 2018后在早期LabVIEW版本中测试这些VI。

- **运行您的测试**—在LabVIEW 2018中打开VI，进行与在早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 2018»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 2018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 2018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 2013或更早版本升级

关于从LabVIEW 2013或更早版本升级至LabVIEW 2018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`upnote14zhs`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 2014升级

从LabVIEW 2014升级到LabVIEW 2018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、从LabVIEW 2016升级和从LabVIEW 2017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 2018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升级和从LabVIEW 2017升级。

在LabVIEW2016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 2018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 2017升级。

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。

从LabVIEW 2017升级

从LabVIEW 2017升级到LabVIEW 2018时可能会遇到下列兼容性问题。

LabVIEW运行时引擎的向后兼容性

从LabVIEW 2017开始，LabVIEW支持LabVIEW运行引擎的向后兼容性。您可以加载并运行旧版LabVIEW中创建的二进制文件和VI，而无需在当前版本的LabVIEW中重新编译VI或重新生成二进制文件。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。

报表生成VI的行为改变

在LabVIEW 2018中，报表生成VI不再支持以标准报表格式生成报表。只能以HTML、Word或Excel格式生成报表。由于行为改变，以下VI已停用：

- **简易打印VI面板或说明信息**—该VI已停用。请改用**打印VI前面板或说明**VI。
- **简易文本报表**—该VI已停用。请改用**创建简易文本报表**VI。
- **获取报表类型**—该VI已停用。请改用**报表类型**VI。
- **新建报表**—该VI已停用。请改用**创建报表**VI。
- **设置报表制表符宽度**—该VI已停用。

不再支持的VI、函数和节点

LabVIEW 2018及更高版本不再支持**数值至枚举**VI。请改用强制转换至类型函数。

LabVIEW 2018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 2018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。

针对不同数据类型自定义自适应VI

比较选板新增检查类型子选板。使用“检查类型”VI和函数可强制让自适应VI(.vim)只接受满足特定要求的数据类型。使用类型专用结构可为指定数据类型自定义自适应VI (.vim)中的代码段。

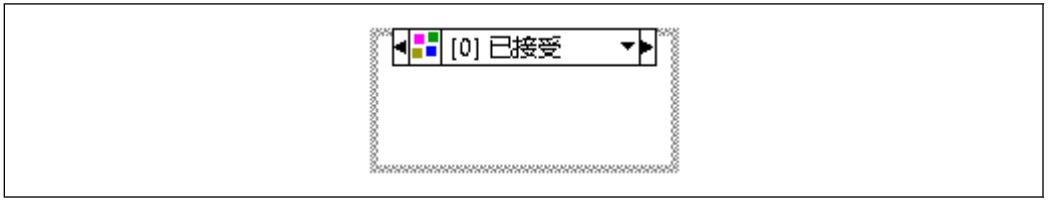

图 2

关于使用类型专用结构自定义自适应VI代码段的范例，见labview\examples\Malleable VIs\Type Specialization Structure\Malleable VIs - Type Specialization Structure.lvproj。

使用用于LabVIEW的命令行接口运行操作

LabVIEW 2018允许您使用用于LabVIEW的命令行接口(CLI)执行命令，在LabVIEW中运行操作。例如，可使用用于LabVIEW的CLI，自动化LabVIEW应用程序的生成过程。用于LabVIEW的CLI支持以下操作：

- MassCompile—批量编译指定目录中的文件。
- ExecuteBuildSpec—使用指定生成规范中的设定生成应用程序、库或比特文件，并返回输出文件的路径。
- RunVI—使用预定义连线板接口运行VI，并返回输出或错误信息。
- CloseLabVIEW—关闭LabVIEW，无提示。
- （**VI Analyzer工具包**） RunVIAnalyzer—LabVIEW VI Analyzer工具包中运行指定的VI分析器任务，并将测试报告保存到指定位置。
- （**Unit Test Framework工具包**） RunUnitTests—在LabVIEW Unit Test Framework工具包中对指定文件运行测试，并将JUnit文件保存到指定位置。

注： 要在LabVIEW中运行此操作，必须使用JKI VI Package Manager(VIPM)软件安装UTF Junit报告库。关于使用VIPM软件的详细信息，请在LabVIEW帮助的**目录**栏中选择**工具包»通过VI Package Manager (VIPM)软件访问LabVIEW附加软件**。

也可在LabVIEW中创建自定义操作。

关于使用用于LabVIEW的CLI的详细信息，请在LabVIEW帮助的**目录**栏中选择**基础»使用用于LabVIEW的命令行接口运行操作**主题。

从LabVIEW调用Python代码

互连接口选板新增Python子选板，可使用它从LabVIEW代码中调用Python代码。Python选板包含以下函数：

- 打开Python会话－用特定版本的Python打开Python会话。
- Python节点－直接调用Python函数。

- 关闭Python会话—关闭Python会话。

注：必须安装Python 2.7或3.6版本以使用LabVIEW Python函数。尽管不支持的Python版本可以与LabVIEW Python函数一起使用，但NI建议仅使用支持的Python版本。访问`ni.com/info`并输入信息代码`python`，了解更多关于安装Python的信息。

应用程序生成器的改进

LabVIEW 2018对LabVIEW应用程序生成器和程序生成规范进行了下列改进。

在Windows和Linux Real-Time终端上创建程序包

您可在LabVIEW中创建程序包，并通过NI Package Manager或SystemLink将软件包分发给用户。您可以使用Package Manager和SystemLink来分发所有类型的文件，包括源代码分发、打包项目库、共享库、.NET程序集和可执行文件。

（**Windows 64位**）创建NI程序包（`.nipkg`）：在**项目浏览器**窗口中右键单击**程序生成规范**并选择**新建»Package**。您的客户可以使用Package Manager或SystemLink订阅源，以查找和安装您的程序包。

（**NI Linux Real-Time**）如安装了LabVIEW Real-Time模块，也可在NI Linux Real-Time终端上创建程序包（`.ipk`）。您的客户可通过SystemLink或通过NI Linux Real-Time终端上的命令行来安装程序包。Package Manager不支持`.ipk`文件。

关于创建程序包的详细信息，请在LabVIEW帮助的**目录**栏中选择**基础»生成和发布应用程序»创建程序生成规范»创建用于发布的程序包**主题。

LabVIEW生成的.NET程序集的向后兼容性支持

通过支持向后兼容性，.NET互操作程序集可加载至创建该程序集的LabVIEW版本中，或机器上安装的最新版本的LabVIEW运行引擎中。例如，您可以在2018版本之后的LabVIEW运行引擎中加载和运行使用LabVIEW 2018构建的.NET互操作程序集，而无需重新编译。

要启用对.NET程序集的向后兼容性支持，请勾选**.NET互操作程序集属性**对话框**高级**页面上的**允许未来版本的LabVIEW加载该.NET程序集**复选框。

对于在LabVIEW 2018及以后版本中创建的程序生成规范，LabVIEW默认启用该选项。您可禁用该选项，将程序生成规范绑定至特定版本的LabVIEW。禁用该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，对话框中不显示该选项，但功能默认为启用。

环境改进

LabVIEW 2018包含以下对LabVIEW环境的改进：

创建自定义类型的功能改进

LabVIEW 2018提供更多创建自定义类型的方式，可将自定义控件的所有实例链接到已保存的自定义控件文件。可通过以下方式新建自定义类型：

- 选择**文件»新建**，然后在**其他文件**下选择**自定义类型**。
- 在**项目浏览器**窗口中右键单击**我的电脑**，从快捷菜单中选择**新建»自定义类型**。

(NI论坛用户Mathis_B提供的建议。)

用于格式化文本的键盘快捷键

在LabVIEW环境中编辑文本时，使用以下键盘快捷键来格式化字体样式：

- <Ctrl-B>—加粗文本。
- <Ctrl-I>—斜体文本。

- <Ctrl-U>—下划线文本。

(NI论坛用户vt92提供的建议。)

程序框图的改进

LabVIEW 2018对程序框图和相关功能进行了以下改进：

并行For循环中错误处理的改进

LabVIEW 2018新增了错误寄存器以简化启用了并行循环的For循环的错误处理。错误寄存器取代了并行For循环上错误簇的移位寄存器，如以下程序框图所示。

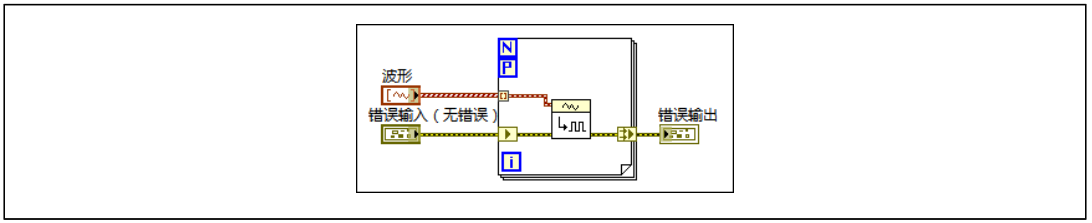

图 3

错误寄存器可自动合并并行循环的错误。在For循环上配置并行循环时，LabVIEW将自动把移位寄存器转换为错误寄存器，从而遵循通过移位寄存器传输错误的最佳实践。

错误寄存器和移位寄存器的运行时行为不同。左侧错误寄存器接线端的行为类似于不启用索引的输入隧道，每个循环产生相同的值。右侧错误寄存器接线端合并每次循环的值，使得来自最早循环的错误或警告值（按索引）为错误寄存器的输出值。如果For循环执行零次，则连接到左侧隧道的值将移动到右侧隧道的输出。

删除并重连对象的改进

当您删除并重连选中的程序框图对象时，LabVIEW也会移除选择矩形中的任何装饰，包括自由标签。在程序框图对象周围拖拽矩形选择框，右键单击选中的对象，并选择**删除并重连**，可删除并重连对象。您还可以在选中对象后，使用**快速放置**键盘快捷键<Ctrl-Space>和<Ctrl-R>键，删除并重新连接对象。

前面板改进

NXG风格控件

控件选板包含新的**NXG风格**前面板控件。使用NXG风格的控件，创建LabVIEW NXG风格的前面板。控件的外观随终端用户运行VI的平台改变。如果将VI迁移到LabVIEW NXG，使用这些控件可以最大限度地减少前面板的失真。

新增VI和函数

LabVIEW 2018中新增了下列VI和函数：

- 比较选板新增检查类型子选板，包含以下VI和函数：
  - 检查数组维数是否一致
  - 检查数组维数大小是否一致
  - 检查是否为复数数值型
  - 检查是否为错误簇型
  - 检查是否为定点数值型
  - 检查是否为浮点数值型
  - 检查是否为小数数值型

  - 检查是否为整型
  - 检查是否为实数浮点数值型
  - 检查是否为实数数值或波形类型
  - 检查是否为实数数值型
  - 检查是否为相同类或子孙类
  - 检查是否为标量数值或波形类型
  - 检查是否为标量数值型
  - 检查是否为有符号整型
  - 检查结构类型是否匹配
  - 检查是否为无符号整型
  - 类型专用结构
- 互连接口选板新增Python子选板，包含以下函数：
  - 打开Python会话
  - Python节点
  - 关闭Python会话
- 转换选板新增强制转换至类型函数。使用该函数可将输入数据转换为兼容的数据类型，同时保留数据值。与强制类型转换函数不同，该函数不会重解析输入数据。在下列场景使用该函数：
  - 消除强制转换点
  - 将不带有类型自定义的数据转换为兼容类型自定义，反之亦然
  - 重命名线上数据（如用户事件引用句柄）

  (NI论坛用户JackDunaway提供的建议。)
- 定时选板新增高精度轮询等待VI。使用该VI可等待指定秒数，其精度高于等待(ms)函数的精度。

新增和改动的属性和方法

LabVIEW 2018包含下列新增和改动的属性和方法：

- LeftShiftRegister类新增“是错误寄存器”属性。使用该属性可读取某个移位寄存器是否为错误寄存器。错误寄存器是一种特殊形式的移位寄存器，它存在于启用了并行循环的For循环中，且其数据类型是错误簇。
- VI类新增“将前面板配置为隐藏顶层”方法。当VI作为顶层VI运行时，使用该方法可隐藏VI的前面板，并可选择在任务栏中隐藏该VI。例如，对于在LabVIEW中创建的独立应用程序，可使用该方法隐藏其启动VI的前面板。
- DisableStructure类新增“禁用样式”属性。使用该属性可读取一个结构是程序框图禁用结构、条件禁用结构还是类型专用结构的信息。
- 更改禁用样式（类：DisableStructure）方法的**禁用样式**参数新增Type Specialization Style选项。使用该选项可将程序框图禁用结构或条件禁用结构更改为类型专用结构。

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 2014升级说明—`upnote14zhs`
- LabVIEW 2015升级说明—`upnote15zhs`
- LabVIEW 2016升级说明—`upnote16zhs`
- LabVIEW 2017升级说明—`upnote17zhs`

371780P-0118

2018 年 3 月