

# LabVIEW™ 2019 升级说明

本文档介绍在 Windows、macOS 和 Linux 操作系统上升级至 LabVIEW 2019 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 2019 的新增功能和行为改动

## 目录

---

升级至 LabVIEW 2019.....	2
推荐升级流程的概述.....	2
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.....	4
4. 转换 VI 并修正 VI 操作上的改动.....	4
常见升级问题疑难解答.....	6
升级和兼容性问题.....	6
从 LabVIEW 2015 升级.....	6
从 LabVIEW 2016 升级.....	6
从 LabVIEW 2017 升级.....	6
从 LabVIEW 2018 升级.....	7
LabVIEW 2019 的功能及改动.....	10
新的群体类型：集合和映射表.....	10
新增的快捷菜单项.....	12
应用程序生成器的改进.....	12
前面板改进.....	13
替换项目项.....	14
高亮代码段的执行.....	15
使用历史探针监视数据流.....	15
在条件结构中处理错误的改进.....	16
指示常量、输入控件和显示控件的文本溢出.....	16
环境改进.....	17
新增和改动的 VI 和函数.....	17
新增属性.....	20
前期版本 LabVIEW 的改动.....	20

# 升级至 LabVIEW 2019

虽然用户可通过安装新版本的 LabVIEW，然后使用最新版本软件加载 VI 来升级小型应用，但 NI 建议用户使用更精准的升级流程，进而有效地检测和校正升级所遇到的困难。



**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、程序版本为不支持的 LabVIEW 版本。关于 NI 仍对其提供主流支持的 LabVIEW 的版本信息，请访问 NI 网站 [ni.com/info](http://ni.com/info) 并输入信息代码 [lifecycle](#) 查询。

## 推荐升级流程的概述

备份VI和机器配置	对开发计算机上的VI采取保护措施，确保能够恢复文件并在必要的情况下可重新升级。
测试和记录VI的现有行为	在之前版本的LabVIEW中验证VI的行为特性，有助于在升级后快速找到VI的行为变化。
安装LabVIEW、附加软件和驱动程序	同时升级全部NI软件，以确保升级后的VI能够在新版本LabVIEW中找到所需的子VI选板以及属性节点。
转换VI并确定VI的行为更改	升级后立即转换并测试VI，确保VI按预期运行并修复更改的行为。



**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](http://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 2019 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 LabVIEW 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。如没有备份文件，则无法在前期版本的 LabVIEW 中打开升级后的 VI，除非将每个 VI 保存为前期版本。

将 VI 提交至源代码控制来备份一组 VI。如升级 VI 后，VI 的行为发生了不可预期的改变，通过本操作可将 VI 恢复为源代码控制中保存的版本。

关于在 LabVIEW 中使用源代码控制的详细信息，见 *LabVIEW 帮助* 的 **基础**»**使用项目和终端**»**概念**»**在 LabVIEW 中使用源代码控制**。

## 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，很难将其恢复为前期版本。因此，如从不支持版本的 LabVIEW 升级或应用程序故障将造成重大损失的情况下，可考虑使用下列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。升级后如要将机器恢复为原始配置，部署备份磁盘镜像即可。
- 在测试机上测试升级过程 - 使用测试机（通常是虚拟机）测试升级过程。尽管在测试机上完成升级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，以避免或最小化故障时间。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

升级 VI 时，LabVIEW 前期版本与 LabVIEW 2019 的差异可能会影响 VI 的行为。在升级前后的 LabVIEW 中测试 VI，并比较升级所产生的行为影响。请确保记录下列测试的当前结果：

- 确定是否存在断开 VI 的批量编译记录

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，见 *LabVIEW 帮助* 的 **基础**»**创建 VI 和子 VI**»**详解**»**保存 VI**»**批量编译 VI**。

- 验证单个 VI 是否正确实现了预期功能的单元测试
- 验证项目或子 VI 组是否按预期运行的集成测试
- 验证 VI 部署到终端（如桌面或 FPGA 终端）后是否正常运行的部署测试
- 检测 CPU 使用量、内存使用量及代码执行速度的性能测试

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



**注：** 测试中如修改了 VI，进行下一步之前请先备份 VI。

关于测试 VI 的详细信息，见 *LabVIEW 帮助* 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](#)。

## 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

要在 LabVIEW 2019 的 VI 中使用通过上一个版本 LabVIEW 创建的自定义控件和 VI，可将上一个版本 LabVIEW 的 labview\user.lib 目录下的文件复制到 LabVIEW 2019 的 labview\user.lib 目录下。

### c. 重新安装 VI 程序包

如使用 JKI VI Package Manager (VIPM) 安装了前期版本 LabVIEW 的 VI 程序包，请打开 VIPM 软件并将全部程序包重新安装为 LabVIEW 2019。

## 4. 转换 VI 并修正 VI 操作上的改动

在 LabVIEW 2019 中批量编译 VI 会将 VI 转换为最新的版本，并创建一个错误记录以帮助用户确定断开的 VI。参考该记录信息与 [升级和兼容性问题](#)，确定并修正版本迁移过程中出现的问题。



**注：** 建议使用 LabVIEW 中的源控制备份 VI 及跟踪更改。如升级 VI 后，VI 的行为发生了不可预期的改变，通过本操作可将 VI 恢复为前期版本。

### a. 在新版 LabVIEW 中批量编译 VI

批量编译 VI 的同时，VI 被转换并保存为 LabVIEW 2019。批量编译 VI 后，前期版本的 LabVIEW 无法打开这些 VI，除非在当前 LabVIEW 中打开每个 VI 或项目，选择 [文件](#)»[保存为前期版本](#)。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，见 *LabVIEW 帮助* 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 VI 使用了更改的功能，前期版本 LabVIEW 和 LabVIEW 2019 间的差异可能会导致某些 VI 出现断线。按照下列步骤快速找到并修复 LabVIEW 2019 VI 中的断线：

1. 将之前生成的批量编译错误记录与测试 VI 当前行为的记录比较，找到升级过程中出现断线的 VI。
2. 要判断是否因 LabVIEW 升级造成 VI 断开，请参考 [升级和兼容性问题](#)。

## c. 找到并修正行为改动

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

- 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测试：
  1. 为全部高于前期版本的 LabVIEW 下载 VI 分析器升级测试。请访问 NI 网站 [ni.com/info](http://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 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](#)，解决测试失败的问题。
- 阅读升级文档
  - [升级和兼容性问题](#) - 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](#) 中提到的已停用对象及其他对象，请打开升级后的 VI 并选择 **编辑»查找和替换**。

- LabVIEW 2019 已知问题列表 - 列出了 LabVIEW 2019 以及之前版本中所有已知程序漏洞。访问 NI 网站 [ni.com/info](http://ni.com/info) 并输入信息代码 [lv2019ki](#)，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 - 列出了特定模块和工具包（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 2019 后在前期 LabVIEW 版本中测试这些 VI。

- 运行测试 - 在 LabVIEW 2019 和前期版本 LabVIEW 中执行相同的 VI 测试，并对结果进行比较。如存在行为改动，请参考升级说明。

# 常见升级问题疑难解答

关于解决以下升级问题的详细信息，请参阅 labview\manuals 目录中安装的 troubleshooting\_guide.html 文档：

- 查找缺失的模块和工具包功能
- 查找缺失的子 VI、选板和属性节点
- 确定 LabVIEW 2019 无法打开在前期版本 LabVIEW 中保存的 VI 的原因
- 确定安装的 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 中可能破坏或改变 VI 行为的改动。

关于新版本 LabVIEW 的已知问题、其他兼容性问题及 LabVIEW 2019 的新功能的详细信息，见 labview 目录中的 readme.html 文件。

### 从 LabVIEW 2015 升级

从 LabVIEW 2015 升级到 LabVIEW 2019 时可能会遇到下列兼容性问题。

在 LabVIEW 2016 及后续版本中，**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 及后续版本的快捷方式列表。

### 从 LabVIEW 2016 升级

从 LabVIEW 2016 升级到 LabVIEW 2019 时可能会遇到下列兼容性问题。

#### 操作者框架 VI 中的行为改动

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 及后续版本中，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 2019 时可能会遇到下列兼容性问题。

#### LabVIEW 运行引擎的向后兼容性

从 LabVIEW 2017 开始，LabVIEW 支持 LabVIEW 运行引擎的向后兼容。即可以加载并运行前期版本 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 后续的版本可加载

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	允许后续版本的 LabVIEW 运行引擎运行该应用程序
打包项目库	打包库属性	允许后续版本的 LabVIEW 加载该打包库
共享库 (DLL)	共享库属性	允许后续版本的 LabVIEW 加载该共享库

对于在 LabVIEW 2017 及后续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更改，并有助于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 的行为改动

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。由于行为改动，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 前面板或说明信息 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

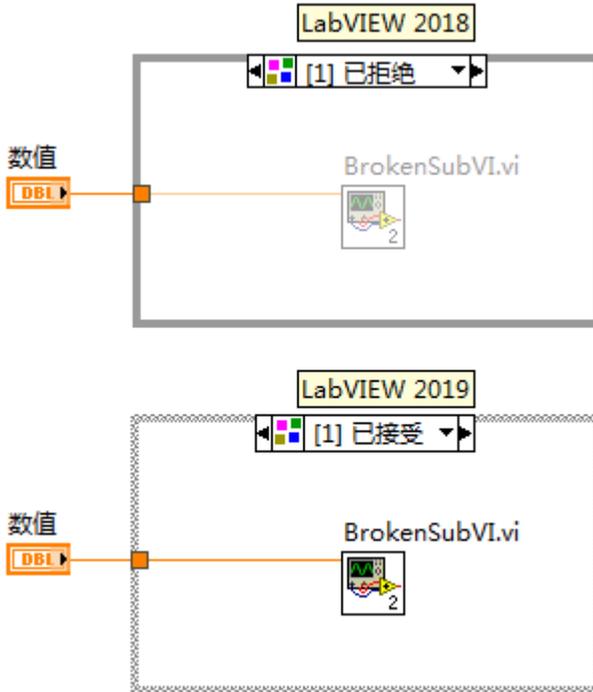
LabVIEW 2018 及后续版本不再支持数值至枚举 VI。请改用强制转换至类型函数。

## 从 LabVIEW 2018 升级

从 LabVIEW 2018 升级到 LabVIEW 2019 时可能会遇到下列兼容性问题。

### 类型专用结构的行为更改

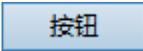
在 LabVIEW 2019 中，类型专用结构在决定接受或拒绝子程序框图时，更改了语法错误检查行为。在 LabVIEW 2018 中，类型专用结构将结构中的错误（如断线）及来自断开的子 VI 或其他依赖项的错误视为拒绝接受子程序框图的原因。在 LabVIEW 2019 中，类型专用结构仅将结构中的错误视为拒绝接受子程序框图的原因。



关于类型专用结构的详细信息，见 *LabVIEW 帮助* 的 **VI 和函数»编程 VI 和函数»结构»类型专用结构** 主题。

### 使用经典、系统及 NXG 风格创建布尔控件的行为更改

当 VI 被配置为使用经典、系统或 NXG 风格创建控件时，LabVIEW 2019 更改了从布尔接线端创建控件的行为。下表比较了 LabVIEW 2018 及更早版本与 LabVIEW 2019 中创建的布尔控件的外观。

样式	类型	LabVIEW 2018 及更早版本	LabVIEW 2019
经典	输入控件		<input checked="" type="checkbox"/> OFF/ON
系统	输入控件/显示控件	<input type="radio"/> OFF/ON	<input checked="" type="checkbox"/> OFF/ON
NXG	输入控件		<input checked="" type="checkbox"/> Off/On



**注：** 如要配置创建输入控件/显示控件的控件样式，单击**文件»VI 属性** 并从**类别**下拉菜单中选择**编辑器选项**，然后在**创建输入控件/显示控件的控件样式**列表中选择样式。

行为更改适用于通过下列方式创建的布尔输入控件和显示控件：

- 通过创建输入控件或创建显示控件方式。
- 右键单击布尔接线端，从快捷菜单中选择**创建»输入控件**或**创建»显示控件**。

#### 数据值引用读取/写入元素（边框节点）的自动错误处理的行为更改

如添加一对数据值引用读取/写入元素边框节点至元素同址操作结构，左右两个边框节点均具有**错误输出**接线端。在 LabVIEW 2018 中，如生成了错误且 VI 已启用自动错误处理，LabVIEW 将为每个未连线的**错误输出**接线端显示错误对话框。在 LabVIEW 2019 中，无论未连线的**错误输出**接线端数量为多少，LabVIEW 均只显示一个错误对话框。

关于数据值引用读取/写入元素（边框节点）的详细信息，见 *LabVIEW 帮助的 VI 和函数»编程 VI 和函数»结构»元素同址操作结构»数据值引用读取/写入元素（边框节点）* 主题。

#### 在受密码保护的库中调用库内成员的行为更改

在 LabVIEW 2018 中，如友元 VI 在受密码保护的库中调用库内成员，首次编辑或运行友元 VI 时必须提供库的密码。在 LabVIEW 2019，用户可编辑或运行友元 VI，而无需提供库的密码。

#### 指示常量、输入控件和显示控件的文本溢出

默认情况下，LabVIEW 2019 指示可见的文本在字符串、数值、时间标识、文本下拉列表/枚举及组合框内被截断。该指示由箭头和文本淡出效果表示。如要禁用文本溢出，可右键单击输入控件、显示控件或常量，取消选择**显示项»文本溢出**。

## 接线端名称更改为数据类型解析 VI

数据类型解析 VI 的顶层接线端名称由大写更改为小写。下一层接线端名称（如簇元素）保持不变。

关于数据类型解析 VI 的详细信息，见 *LabVIEW 帮助的 VI 和函数* » *编程 VI 和函数* » *簇、类、变体 VI 和函数* » *变体 VI 和函数* » *数据类型解析 VI* 章节。

# LabVIEW 2019 的功能及改动

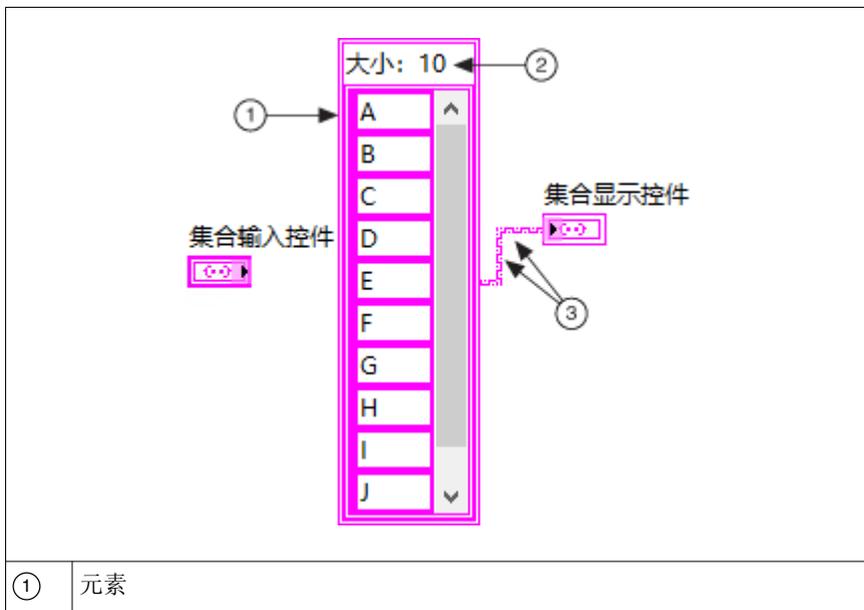
观点交流符号  表示来自 ni.com 上 [NI 观点交流](#) 论坛的产品反馈建议。

有关不同版本 LabVIEW 的升级和兼容性问题，见 [升级和兼容性问题](#)。关于 LabVIEW 2019 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，请参阅 labview\readme 目录中的 readme.html 文件。

## 新的群体类型：集合和映射表

LabVIEW 2019 引入了下列用于聚合同类数据的 *群体数据类型*：*集合*和*映射表*。两个类型均依照排序维护具有唯一值的元素或条目。相比未排序的数据结构（如数组），群体能够对数据进行更快地搜索、插入、更改及移除操作，并对于较大的数据同样适用。

- **集合**—具有相同数据类型及唯一值的元素的集合。



②	大小 - 集合中的元素数量
③	集合连线 - 连线颜色与元素的数据类型颜色匹配

- **映射表**—由键-值对组成的条目集合，其中所有键都是唯一的。键和值可分别为任意数据类型。映射表也称为字典，因为键用于查找值。

①	键
②	值
③	条目 - 关联的键-值对
④	大小 - 映射表中的输入数量
⑤	映射表连线 - 连线颜色与值的数据类型颜色匹配

使用集合和映射表 VI 和函数（位于**编程**»**群体** 选板）及集合和映射表输入控件和显示控件（位于**数据容器**选板）创建/操作集合和映射表。

**注：** 不能通过交互式或编程方式修改集合或映射表中的嵌入数据。仅能通过下列操作来整体更新集合或映射表数据：写入集合或映射表前面板接线端；或右键单击集合或映射表控件并选择**数据操作**»**复制数据**或**数据操作**»**粘贴数据**。

关于集合和映射表的详细信息，见 *LabVIEW 帮助*的**基础**»**用字符串、簇、数组和群体将数据分组**»**概念**»**用群体将数据分组**主题。

关于集合与映射表 VI 和函数的详细信息，见 *LabVIEW 帮助的 VI 和函数*»*编程 VI 和函数*»*群体 VI 和函数*。

关于使用集合或映射表操作数据的范例，见下列 VI 和项目。

- labview\examples\Collections\Set Collection - Word Counting.vi
- labview\examples\Collections\Map Collection - Comparing Test Results.vi
- labview\examples\Collections\Map Collection - Word Counting.vi
- labview\examples\Design Patterns\Registration Map\Registration Map Usage.lvproj

## 新增的快捷菜单项

LabVIEW 2019 包括下列新增的快捷菜单项。

- **创建常量、创建输入控件及创建显示控件**—在接线端上创建常量、输入控件及显示控件。这些快捷菜单项早就存在，本版本中这三项集中位于快捷菜单的顶部。
-  **替换转换**—右键单击数值转换函数并选择**替换转换**，将某一数值转换函数替换为其他数值转换函数。*[NI 论坛用户 EthanStern 提供的建议。]*
-  **创建颜色盒**—右键单击颜色接线端并选择**创建常量、创建输入控件或创建显示控件**，以创建颜色盒常量、输入控件及显示控件。*[NI 论坛用户 altenbach 提供的建议。]*
-  **创建所选项的簇**—右键单击选中的常量、输入控件和显示控件，选择**创建**»**所选项的簇**来创建所选项的簇。*[NI 论坛用户 okubik 提供的建议。]*
-  **创建标量常量或输入控件**—右键单击数组接线端，可选择**创建**»**标量**»**常量**或**创建**»**标量**»**输入控件**来创建标量常量或输入控件。*[NI 论坛用户 altenbach 提供的建议。]*
- **替换比较**—右键单击比较函数并选择**替换比较**，将某一比较函数替换为其他比较函数。

## 应用程序生成器的改进

LabVIEW 2019 对 LabVIEW 应用程序生成器和程序生成规范进行了下列改进。

(Windows) 将应用程序发布为程序包安装程序

创建程序包安装程序 (.exe) 后，可通过 NI Package Manager 将应用程序发布至客户端。*程序包安装程序*包含了全部程序包依赖关系，即客户端可在无网络连接的情况下安装全部程序包。

右键单击*程序生成规范*选择**新建**»**程序包**，然后选择**程序包属性**对话框上**程序包安装程序**页面的**创建程序包安装程序**，可创建程序包安装程序。

发布 NI 程序包至订阅源

可发布 NI 程序包至订阅源，客户可订阅该源以在程序包更新时收到通知，继而通过网络安装 NI 程序包。

**订阅源(Feed)**是一组程序包文件和一个列出程序包信息的 manifest 文件。如要发布互相关有依赖关系的多个 NI 程序包，NI 建议使用订阅源。通过订阅源，用户仅需创建一个用作主源的位置，并通过该源位置维护用于终端用户访问的全部 NI 程序包。

在 LabVIEW 2019 中，可使用**程序包属性**对话框的**订阅源**页面添加 NI 程序包至本地订阅源，或发布 NI 程序包至 SystemLink 订阅源。

- **添加至订阅源** - 将 NI 程序包添加至本地订阅源。客户端可使用 NI Package Manager 订阅所需的源以安装 NI 程序包。
- **发布至 SystemLink 订阅源** - 将在 LabVIEW 中创建的程序包发布至 SystemLink Server 订阅源。客户端可在 SystemLink 中查找和安装 NI 程序包。

关于程序包、程序包安装程序及订阅源的概念性信息，见 *LabVIEW 帮助*的**基础»生成和发布应用程序»创建程序生成规范»发布包类型**主题。

关于创建程序包、程序包安装程序及将它们发布到客户端的详细信息，见 *LabVIEW 帮助*的**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**章节。

### 其他应用程序生成器的改进

**程序包属性**对话框的**程序包属性**页面被重命名为**程序包**。**程序包**页面包含下列新增选项：

- **包括 NI 证书包** - 指定包含 NI 证书包 (ni-certificates)。客户端在 NI Package Manager 中安装应用程序时将 NI 证书包作为推荐的依赖关系。
- **在运行引擎部署包列表中显示** - 在**筛选选项**中选择**仅运行引擎部署包**时，指定程序包显示在**依赖关系**页上的**相关程序包**列表中。
- **按产品筛选时在 NI Package Manager 中显示** - 在 NI Package Manager 中选择**仅产品**来筛选程序包时，指定显示该程序包。如创建的程序包是要包含在 NI Package Manager 的**浏览产品**部分中的产品，请启用该选项。

## 前面板改进

LabVIEW 2019 包含以下对前面板的改进：

### 集合和映射表输入控件和显示控件

在 LabVIEW 2019 中，**数组、矩阵与簇**选板被重命名为**数据容器**。该选板包含新增的映射表和集合控件。

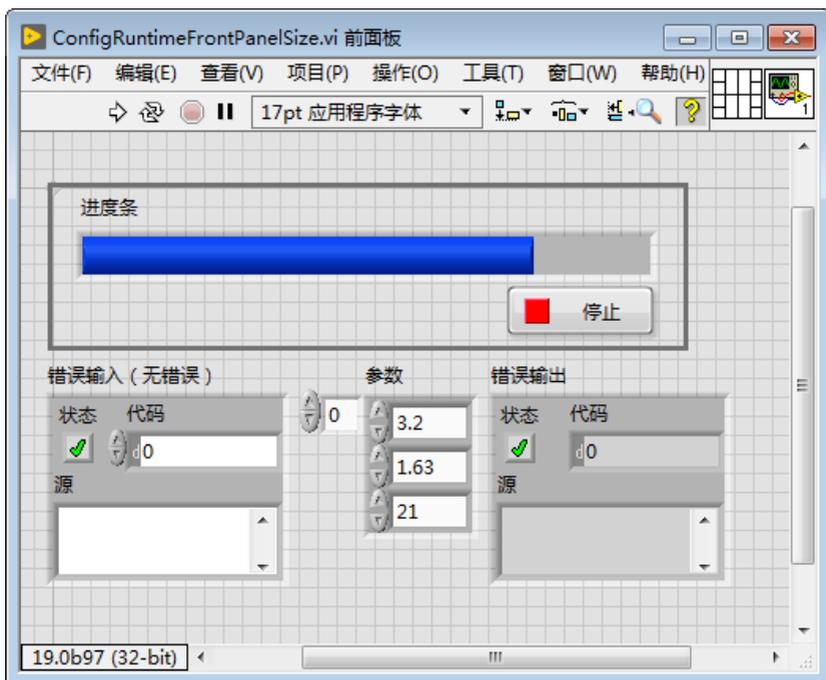
关于数据容器输入控件和显示控件的详细信息，见 *LabVIEW 帮助*的**基础»创建前面板»概念»前面板控件»数据容器控件**主题。

### 更多的 NXG 风格控件

LabVIEW 2019 在 NXG 风格选板上包含了更多的输入控件和显示控件，您可以轻松创建与 LabVIEW NXG 视觉风格相同的数值、布尔、容器和 I/O 名称输入控件和显示控件。

## 编辑时查看运行时前面板边界

在 LabVIEW 2019 中，可配置 VI 记住运行时前面板大小和滚屏位置。当 VI 处于编辑模式时，LabVIEW 将设置映射为前面板上的矩形边框。VI 运行时，矩形边框消失。在下图中，矩形边框指示了 VI 运行时前面板可见部分的位置和大小。



关于查看运行时前面板边界的说明，见 *LabVIEW 帮助的基础»创建前面板»详解»设计用户界面»编辑时查看运行时前面板边界* 主题。



**注：** LabVIEW 仅显示单窗格前面板的运行时前面板边界。

[NI 论坛用户 *altenbach* 提供的建议。]

## 替换项目项

在 LabVIEW 2019 中，**替换为打包库** 快捷菜单被重命名为 **替换为**。可将 LabVIEW 项目中的 VI、类或库替换为相同类型的其他项。在 **项目浏览器** 窗口右键单击项目项，选择 **替换为** 以选择要替换原始项的文件。

关于替换项目项的详细信息，见 *LabVIEW 帮助的基础»使用项目和终端»概念»在 LabVIEW 中管理项目»替换项目项* 主题。

## 高亮代码段的执行

LabVIEW 2019 允许用户高亮代码段的执行，而无需高亮执行全部 VI 程序框图代码。右键单击连线并选择**自定义探针**»**切换高亮显示执行过程**，可设置要切换高亮显示执行过程的点。VI 运行时，探针在指定的点切换高亮显示执行过程。

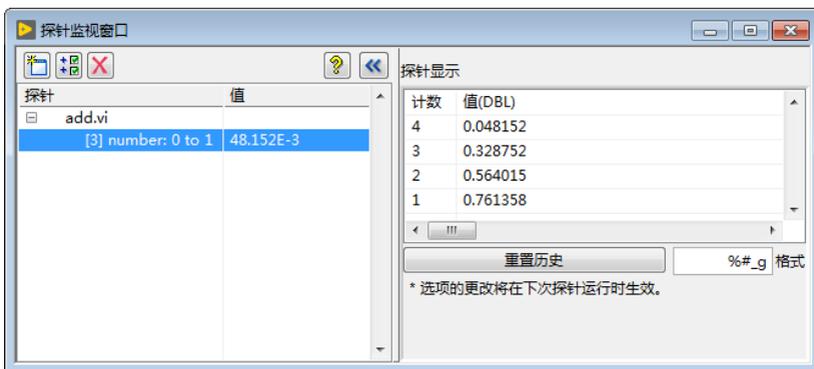


**注：** RT 终端不支持**切换高亮显示执行过程**自定义探针。

[NI 论坛用户 therealkilkenny 提供的建议。]

## 使用历史探针监视数据流

在 LabVIEW 2019 中，可在连线上放置历史探针来显示流经连线的历史数据。



历史探针适用于下列数据类型：

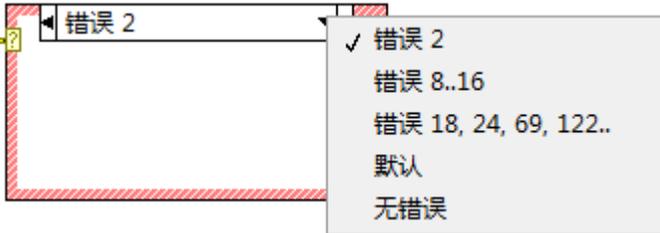
- 布尔
- 字符串
- 路径
- I8
- I16
- I32
- I64
- U8
- U16
- U32
- U64
- SGL
- DBL
- EXT

- CSG
- CDB
- CXT

要在连线上放置历史探针，请右键单击连线并选择**自定义探针**，然后选择用户数据类型的历史探针。

## 在条件结构中处理错误的改进

在 LabVIEW 2019 中，可在选择器标签中配置条件结构为：执行指定错误的子程序框图，或执行通过错误值或范围指定的一系列错误的子程序框图。该错误值和范围的语法与输入数字值或范围的语法相同。

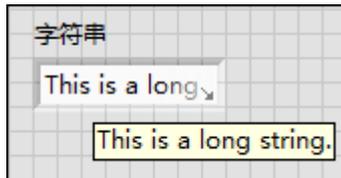


[NI 论坛用户 Hueter 提供的建议。]

关于在条件结构中处理错误的详细信息，见 *LabVIEW 帮助* 的 **基础** » **运行和调试 VI** » **详解** » **错误检查和处理** » **使用条件结构处理错误** 主题。

## 指示常量、输入控件和显示控件的文本溢出

LabVIEW 2019 通过文本淡出效果与箭头的配合，指示可见的文本在字符串、数值、时间标识、文本下拉列表/枚举及组合框内被截断。将鼠标悬停在箭头上可在一个提示条内显示全部文本信息。



LabVIEW 2019 默认启用文本溢出。如要禁用文本溢出，可右键单击输入控件、显示控件或常量，取消选择**显示项** » **文本溢出**。

[NI 论坛用户 Lavezza 提供的建议。]

# 环境改进

LabVIEW 2019 包含以下对 LabVIEW 环境的改进：

-  双击动态分配子 VI 节点时，**选择实现**对话框包含新增的**程序框图预览**。该预览将显示选中 VI 的程序框图预览。[NI 论坛用户 PrimaryKey 提供的建议。]
- (Windows) 在 LabVIEW 中生成安装程序时，可执行文件的默认名称由 setup.exe 更改为 install.exe。该行为更改可能会对依赖 setup.exe 名称的生成后进程产生潜在影响。如要为可执行文件指定其他名称，可对**安装程序属性**对话框的**产品信息**页上的**安装程序名称**选项进行修改。
- 单击错误下拉列表箭头时，**选择错误**对话框包含新增的**过滤**选项，该选项可用于在指定错误代码范围内搜索关键字。使用此选项可缩小**错误列表**中显示的错误代码列表范围。
-  可在**属性**对话框的**外观**页上设置对象在前面板或程序框图上的左坐标和上坐标位置。[NI 论坛用户 smmarlow 提供的建议。]
- 使用**快速放置**键盘快捷键<Ctrl-F>调整前面板和程序框图的对象及窗口位置。相继按下<Ctrl-Shift-Space>、<Ctrl-F>，LabVIEW 将执行下列整理操作：



**注：** 也可按下<Ctrl-U>整理前面板和程序框图上的对象。

- 在前面板上，将输入控件和显示控件排列为与连线板一致、调整大小、并将其移动到主监视器的左上方位置。
- 在程序框图上，将程序框图滚动到相对于最顶层和/或最左侧的程序框图对象的合理位置、调整大小、并将其移动到主监视器的左上方位置。

## 新增和改动的 VI 和函数

LabVIEW 2019 中新增了下列 VI 和函数：关于 VI、函数和节点的详细信息，见 *LabVIEW 帮助* 的 **VI 和函数**。

### 新增 VI 和函数

#### 映射表、集合 VI 和函数

编程选板中包含了新增的群体子选板。该子选板包含了下列映射表和集合 VI 与函数。

- 映射表 VI 和函数 - 创建和操作映射表。
  - 生成映射表
  - 群体大小
  - 映射表至数组转换
  - 空群体？
  - 映射表同址访问
  - 插入映射表
  - 在映射表中查找

- 映射表常量
- 读取映射表最大值和最小值
- 从映射表中删除
- 注册映射表 VI
  - 确认注册
  - 注册
  - 注销
- 集合 VI 和函数 - 创建和操作集合。
  - 生成集合
  - 群体大小
  - 集合至数组转换
  - 空群体？
  - 插入集合
  - 集合的元素？
  - 读取集合最大值和最小值
  - 从集合中删除
  - 集合的笛卡尔积
  - 集合常量
  - 集合的差集
  - 集合的交集
  - 集合的对称差
  - 集合的并集

## JSONtext VI

平化/还原字符串选板包含安装 JSONtext VI 的链接，该 VI 用于在 LabVIEW 中编程 JSON。在平化/还原字符串选板上单击**安装 JSONtext 附加软件**，通过 JKI VI Package Manager (VIPM) 安装 JSONtext 工具包。JSONtext VI 将出现在**附加软件»JSONtext** 选板中。

*[ 特别感谢 JSONtext 附加软件的作者 James David Powell 博士。 ]*

## 数据类型解析 VI

数据类型解析选板包含下列新增 VI:

- 获取映射表群体信息 - 从存储在输入变量数据中的数据类型获取映射表信息。
- 获取集合群体信息 - 从存储在输入变量数据中的数据类型中获取集合信息。
- 为错误簇 - 指示输入变体中存储的数据类型是否为错误簇。

## 高级文件 VI

高级文件函数选板包含下列新增 VI:

- 递归创建目录 - 创建目录及其所需的父目录。
- 创建文件和包含的文件夹 - 创建位于指定路径下的文件以及该路径中不存在的文件夹。

## 数值 VI 和函数

数值选板中包含下列新增常量和 VI:

-  非法数字 - 返回值 NaN (非法数字) 至程序框图。[NI 论坛用户 *altenbach* 提供的建议。]
- 随机数 (范围) - 从指定范围内生成随机值。该 VI 可以与下列数据类型配合使用: U64、I64 及 DBL。

## 附加字符串 VI 和函数

附加字符串选板中包含下列新增 VI:

- 一维字符串数组至分隔符字符串转换 - 将一维字符串数组的元素转换为单个字符串, 通过分隔符分隔输入数组的元素。
- 分隔符字符串至一维字符串数组转换 - 将带分隔符的字符串中的子字符串转换为一维字符串数组的元素。

## 其他新增 VI 和函数

LabVIEW 2019 中新增了下列 VI 和函数:

- 数组选板包含新增的删除一维数组重复项 VI。该 VI 删除一维数组中的重复元素。该 VI 保留了数组中元素的原始顺序。
- 检查类型选板包含新增的检查结构类型是否不匹配函数。如果输入类型与任何指定的不匹配输入的数据类型相同, 则忽略类型定义和类型名称, 该函数会中断调用 VI。使用检查结构类型是否不匹配函数和类型专用结构自定义自适应 VI (.vim) 中的代码为指定的数据类型, 或强制自适应 VI 拒绝指定的数据类型。
- 协议选板包含新增的等待已配置的网络 VI。该 VI 等待直至系统可连接远程主机。
- 同步选板包含新增的同步数据流 VI。上游代码执行后, 该 VI 传递输入端的值。该 VI 用于在数据流的单个点同步多个并行代码路径, 以确保特定的执行顺序。
- 应用程序控制选板包含新增的获取命令行参数 VI。打开 LabVIEW 或通过 LabVIEW 生成的应用程序时, 该 VI 返回从命令行传递的用户定义参数。用户定义的参数以两个连字符(-- ) 作为开始, 且在命令行中由空格分隔。

## 更改的 VI 和函数

LabVIEW 2019 中包含下列更改的 VI 和函数:

- 位于 **互连接口»Python** 选板的 Python 节点支持新的数据类型 - 布尔。可使用 Python 节点转换数值数组为 NumPy 数组。

# 新增属性

LabVIEW 2019 包含下列新增属性：

- VI 类包含新增的**程序框图窗口:对齐网格大小** 和 **前面板窗口:对齐网格大小**属性。通过上述属性在 VI 的程序框图或前面板上读取/写入对齐网格大小。必须先启用 VI 脚本以使用上述属性。

## 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](http://ni.com/info)，输入下列信息代码：

- LabVIEW 2015 升级说明 - [upnote15zhs](#)
- LabVIEW 2016 升级说明 - [upnote16zhs](#)
- LabVIEW 2017 升级说明 - [upnote17zhs](#)
- LabVIEW 2018 升级说明 - [upnote18zhs](#)

信息如有变更，恕不另行通知。关于 NI 商标的详细信息，请访问 [ni.com/trademarks](http://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](http://ni.com/patents) 上的 *National Instruments Patent Notice*。可在 NI 产品的自述文件中找到最终用户许可协议 (EULA) 和第三方法律声明。请查阅 [ni.com/legal/export-compliance](http://ni.com/legal/export-compliance) 上的 *Export Compliance Information* 以了解 NI 全球出口管制政策，以及如何获取相关的 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。

© 1998—2019 National Instruments. 版权所有

371780R-0118 2019 年 3 月 7 日