

LabVIEW™ 2020 升级说明

本文档介绍在 Windows、macOS 和 Linux 操作系统上升级至 LabVIEW 2020 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 2020 的新增功能和行为改动

目录

升级至 LabVIEW 2020.....	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 2016 升级.....	6
从 LabVIEW 2017 升级.....	6
从 LabVIEW 2018 升级.....	7
从 LabVIEW 2019 升级.....	10
LabVIEW 2020 的功能及改动.....	10
使用 LabVIEW 接口提高代码的灵活性.....	10
LabVIEW Web 服务的改进.....	12
应用程序生成器的改进.....	13
环境改进.....	13
导入共享库向导的改进.....	16
程序框图的改进.....	16
新增和改动的 VI 和函数.....	18
新增和改动的属性、方法和事件.....	20
前期版本 LabVIEW 的改动.....	22

升级至 LabVIEW 2020

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 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，输入信息代码 [upgradeOld](#) 查询。

1. 备份 VI 和机器配置

升级至 LabVIEW 2020 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 2020 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 2020 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 2020 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 2020。

4. 转换 VI 并修正 VI 操作上的改动

在 LabVIEW 2020 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 2020。批量编译 VI 后，之前版本的 LabVIEW 将无法打开这些 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 2020 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 2020 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 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 2020 已知问题列表 - 列出了 LabVIEW 2020 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 [lv2020ki](#)，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。要查看驱动程序的自述文件，请访问 ni.com/manuals。



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 2020 后在前期 LabVIEW 版本中测试这些 VI。

- 运行测试 - 在 LabVIEW 2020 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 2020 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 2020 的新功能的详细信息，见 labview 目录中的 readme.html 文件。

从 LabVIEW 2016 升级

从 LabVIEW 2016 升级到 LabVIEW 2020 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 2020 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。

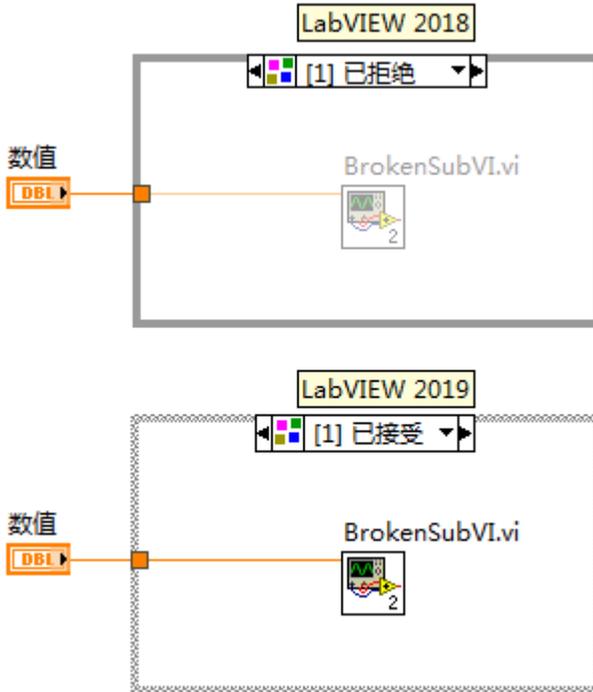
从 LabVIEW 2018 升级

从 LabVIEW 2018 升级到 LabVIEW 2020 时可能会遇到下列兼容性问题。

- 程序包属性对话框的**程序包属性**页面被重命名为**程序包**。
- 数组、矩阵与簇选板被重命名为数据容器。
- 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。

类型专用结构的行为更改

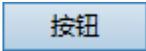
在 LabVIEW 2019 中，类型专用结构在决定接受或拒绝子程序框图时，更改了语法错误检查行为。在 LabVIEW 2018 及更早版本中，类型专用结构将结构中的错误（如断线）及来自断开的子 VI 或其他依赖项的错误视为语法错误。在 LabVIEW 2019 及更高版本中，类型专用结构仅将结构中的错误视为语法错误。



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 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 及更新版本中创建的布尔控件的外观。

样式	类型	LabVIEW 2018 及更早版本	LabVIEW 2019 或更高版本
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 及更新版本中，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 及更新版本中，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 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。要在 LabVIEW 2018 及更早版本中启用文本溢出，可右键单击输入控件、显示控件或簇、簇数组内的常量，并选择**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。

从 LabVIEW 2019 升级

从 LabVIEW 2019 升级到 LabVIEW 2020 时可能会遇到下列兼容性问题。

不再建议使用 MathScript 功能

不再建议在新设计中使用 LabVIEW MathScript。请访问 ni.com/migratemathscript-zh-cn 获取详细信息和推荐的替代方案。

远程前面板的 SSL 加密功能已停用

在 LabVIEW 2020 中，即使勾选了选项对话框 **Web 服务器** 页面 **远程面板服务器** 部分的 **SSL 复选框**，也不再支持安全套接层 (SSL) 加密。如果尝试与远程前面板服务器建立安全连接，LabVIEW 会返回错误。

不再支持的 VI、函数和节点

LabVIEW 2020 及后续版本不再支持 MD5 校验和文件 VI。使用文件校验和 VI。

调用父方法节点的名称改动

调用父方法节点更名为调用父类方法。

LabVIEW 2020 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。

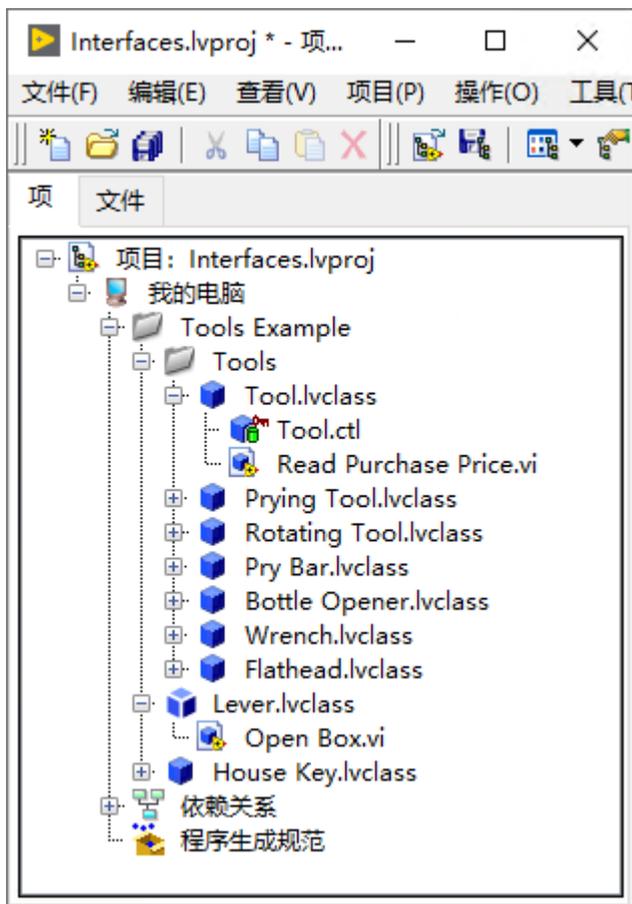
有关不同版本 LabVIEW 的升级和兼容性问题，见 *升级和兼容性问题*。关于 LabVIEW 2020 的已知问题、部分已修正问题、其他兼容性和新增功能的相关信息，请参阅 labview\readme 目录中的 readme.html 文件。

使用 LabVIEW 接口提高代码的灵活性

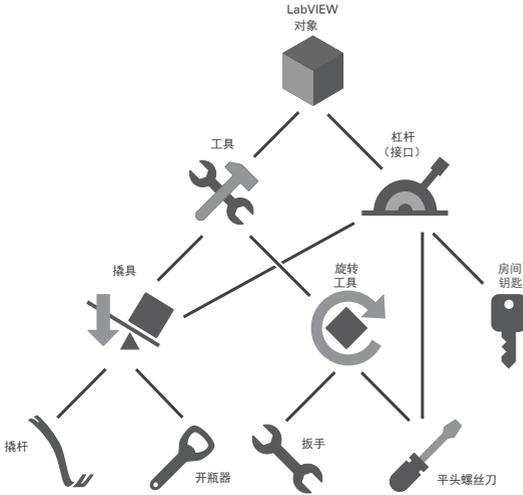
LabVIEW 2020 新增了接口。接口可被认为是不带有私有数据控件的类，但这种微小的差异使得接口在软件架构中可以实现与类完全不同的目的。具体而言即接口可支持多重继承形式。

接口声明了对象可以扮演的角色，而无需定义如何执行该角色。通过继承接口，类声明其对象实现该角色，然后该类负责指定行为的执行方式。当一个类继承了多个接口时，其对象可被传输至需要完成不同角色的多个软件模块中。

以下项目包括工具(Tool)类，以及几个不同工具的子类。该项目还包括杠杆(Lever)接口。可以看到 Tool 类有一个控件，该控件定义了该类中的数据，而 Lever 接口则没有控件，因为接口没有私有数据。类的符号是一个实心的立方体(📦)。接口的符号是立方体的若干面(📦)。接口和类使用相同的文件扩展名 .lvclass。



下图显示了 Tool 类、House Key 类和 Lever 接口之间的继承关系。Tool 类和 Lever 接口均继承自 LabVIEW 对象。除了拥有自己的方法以及它们各自的父类的方法之外，撬具类(Prying Tool)和平头螺丝刀类(Flathead)还继承自杠杆接口(Lever)，因为它们都使用了杠杆原理。继承了杠杆接口(Lever)的方法。由于杠杆(Lever)是接口而不是类，因此这种多重继承是合法的。除 LabVIEW 对象外，三个不相关的类(撬具、平头螺丝刀、房间钥匙)具有一个共同的祖先——杠杆。



可通过下列方式创建接口：

- 在**项目浏览器**窗口，右键单击**我的电脑**并从快捷菜单中选择 **新建**»**接口**。
- 从**新建**列表中选择 **文件**»**新建**，再选择**其他文件**»**接口**。
- 使用 LabVIEW 类:创建接口方法。

关于接口的属性和方法，请参考**新增和改动的属性、方法和事件**。

关于使用接口的范例，请参考下列项目：

- labview\examples\Object-Oriented Programming\Basic Interfaces\Basic Interfaces.lvproj
- labview\examples\Object-Oriented Programming\Actors and Interfaces\Actors and Interfaces.lvproj

LabVIEW Web 服务的改进

LabVIEW 2020 新增了开发 Web 服务并将服务发布至 NI Web 服务器的功能。LabVIEW 2020 还将继续支持应用程序 Web 服务器。

NI Web 服务器是生产级的 Web 服务器，可以托管用户授权的 Web 服务。例如，LabVIEW Web 服务、SystemLink 服务。NI Web 服务器保护 Web 应用程序免受常见 Web 安全威胁，为许多企业级数据服务提供良好的扩展性，可以实现设备管理。

 **注：** NI Web 服务器仅支持 Windows 64 位。应用程序 Web 服务器支持 Windows 32 位和 64 位和 RT 终端。

您可以通过独立应用程序、程序包、程序包安装程序将 Web 服务发布至 NI Web 服务器。可在 NI Web 服务器上启用 SSL 或为用户角色分配不同的权限，在 Web 客户端和 LabVIEW Web 服务之间建立安全的通信。

Web 客户端的默认响应格式从 XML 改为 JSON 字符串。如要创建用户界面，在浏览器中可视化数据并与数据交互，可将 WebVI 集成至 Web 服务中。

关于 WebVI 的更多信息，见 [LabVIEW NXG Web 模块手册](#)。

关于开发、托管和发布 LabVIEW Web 服务的详细信息，见 [LabVIEW 帮助](#) 中的 [基础»传输数据»通过网络传输数据»从 Web 客户端与 LabVIEW 应用程序通信»概述：通过 Web 与 LabVIEW 应用程序通信](#) 主题。

应用程序生成器的改进

LabVIEW 2020 对 LabVIEW 应用程序生成器和程序生成规范进行了下列改进。

(Windows) 修复程序包

如安装没有正确运行，可在 NI Package Manager 中修复程序包。在 LabVIEW 中，当 NI Package Manager 修复程序包时，可配置执行一个自定义的修复操作。要创建一个自定义的修复操作，在 **程序包属性** 对话框的 **高级** 页，右键单击 **自定义操作** 表格，选择 **添加修复操作**。可从下列选项选择：

- **修复前** - 指定修复当前程序包之前要执行的操作。
- **修复后** - 指定修复当前程序包之后要执行的操作。
- **所有修复后** - 指定修复所有程序包之后要执行的操作。

其他应用程序生成器的改进

程序包属性 对话框的 **订阅源** 页有下列新增选项：

- **包括订阅源的所有依赖项** - 指定是否添加程序包的全部依赖关系至订阅源，或仅添加程序包至订阅源。
- **添加程序包版本至订阅源位置** - 指定是否在订阅源中包含程序包版本。LabVIEW 将带有程序包版本的子文件夹附加至订阅源路径。
- **包括 SystemLink 订阅源的所有依赖项** - 指定将程序包的所有依赖项或者程序包本身包括在 SystemLink 订阅源中。

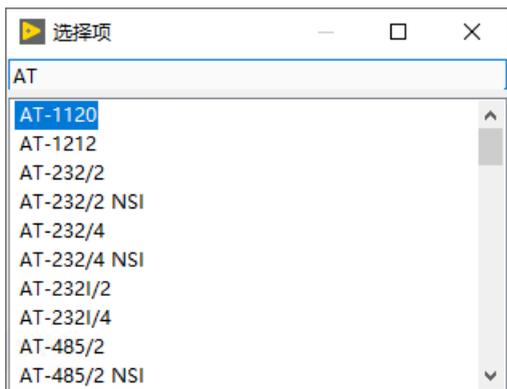
这些选项仅适用于 Windows。

环境改进

LabVIEW 2020 包含以下对 LabVIEW 环境的改进：

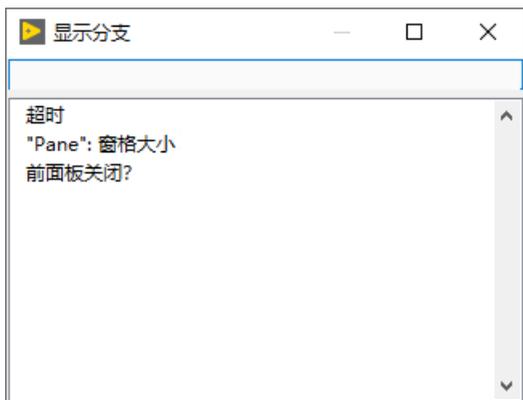
选择列表项的改进

在 LabVIEW 2020 中，可使用 **选择项** 对话框改变下列列表控件、枚举、I/O 控件或常量的列表项的值。右键单击前面板控件/常量或程序框图上的常量，并选择 **选择项**。该对话框取代了之前的 **选择项** 快捷菜单项。



在子程序框图之间切换的改进

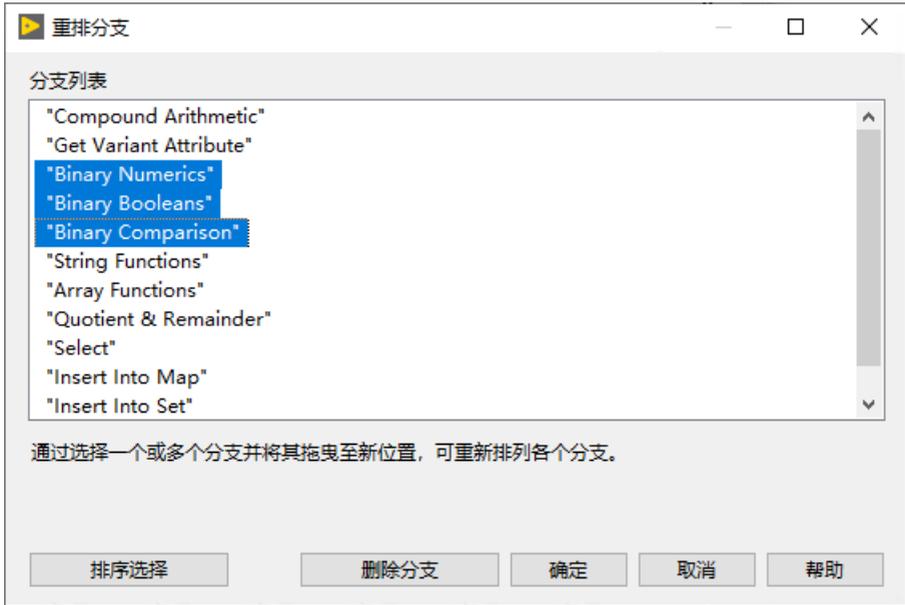
在 LabVIEW 2020 中，可使用**显示分支**对话框改变条件结构、事件结构、条件禁用结构的可见帧。右键单击结构边框，选择**显示分支**。该对话框取代了之前的**显示分支**快捷菜单项。



重排分支或子程序框图的改进

 **重排分支**对话框经过重新设计后新增了下列功能：

- 重排多个分支
- 删除一个或多个分支
- 对分支列表中选中的分支排序
- 调整对话框大小



[NI 论坛用户 Intaris 提供的建议。]

清除变体数据类型的数据

要清除一个变体类型的数据，右键单击变体控件，选择**数据操作»清除数据**。

[NI 论坛用户 altenbach 提供的建议。]

显示错误的改进

在 LabVIEW 2020 中，VI 或库本身没有错误但是有断开的依赖项，VI 和库报告错误的方式有改进。在 LabVIEW 2019 及更早版本中，**错误列表**窗口显示 VI 或库的直接依赖项。双击错误可打开直接依赖项。在 LabVIEW 2020 中，**错误列表**将显示**根本原因：依赖项断开**。双击错误，打开导致错误的断开的依赖项。

打开簇大小对话框

除了右键单击，也可双击簇至数组转换函数，打开**簇大小**对话框。

[NI 论坛用户 PalanivelThiruvankadam 提供的建议。]

显示下拉列表中的最后一项

可隐藏下拉列表中的最后一项，使用户在 VI 运行时无法选择该项。要显示或隐藏最后一项，右键单击下拉列表控件，选择**运行时显示/隐藏最后一项**。也可使用**最后值隐藏**属性，通过程序显示或隐藏下拉列表的最后一项。

对话框的改动

LabVIEW 2020 新增和改进了配置 LabVIEW Web 服务的对话框。

- **Web 服务属性**对话框包含下列改动：
 - **NI Web 服务器权限** - 使用该页面创建和定义 Web 服务的权限。
 - **HTTP 方法 VI 设置** - 使用新增的 **NI Web 服务器**选项卡配置调用 HTTP 方法 VI 的权限。**安全**选项卡更名为**应用程序 Web 服务器**，在该选项卡上可配置与应用程序 Web 服务器相关的权限。
- **程序包属性**对话框新增了 **Web 服务**页，用来将 Web 服务添加至特定的程序包。
- **应用程序属性**对话框的 **Web 服务**页新增了 **NI Web 服务器**选项。可指定 NI Web 服务器托管独立应用程序包含的 Web 服务。
- LabVIEW 2020 重新排列了**选项**对话框 **Web 服务器**页的选项，用于配置访问 Web 服务的方式。Web 服务可托管于 NI Web 服务器，也可托管于应用程序 Web 服务器。

类属性对话框的**继承**页包含下列改动：

- 新增的**父接口**部分 - 定义类从哪个接口继承。
- **继承层次结构**更名为**父类层次结构**。
- **更改继承**选项更名为**更改父类**。

导入共享库向导的改进

在导入共享库向导的**配置 VI 和控件**页，有下列新选项可将设置应用到多个参数和函数：

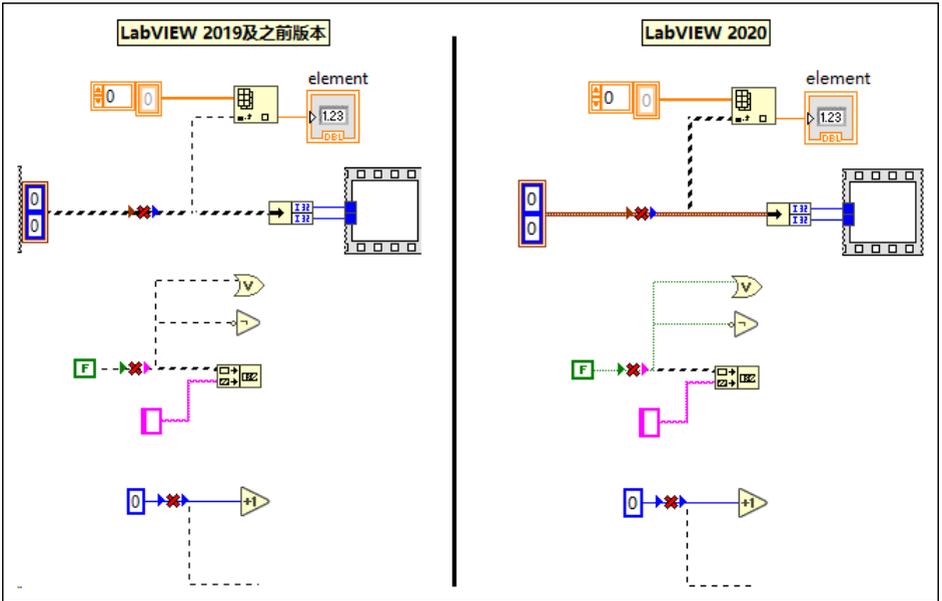
- **应用至全部匹配的参数** - 将当前参数的类型设置应用到与 header 文件中声明匹配的所有函数参数上。
- **应用至全部** - 将当前函数的调用规范应用至共享库的所有函数。

程序框图的改进

LabVIEW 2020 对程序框图和相关功能进行了以下改进：

清理断线的分支

 在 LabVIEW 的早期版本中，如果一根连线上出现断线分支，则整个连线显示为断开。移除断线的分支会同时移除整个连线。在 LabVIEW 2020 中，只有断线的分支显示为断线。选择**编辑**，**移除断线**或按下<Ctrl-B>，LabVIEW 仅移除断线的分支。如果连线的**所有分支均断开**，LabVIEW 将移除整条连线。



[NI 论坛用户 *altenbach* 提供的建议。]

隐藏事件数据节点

在事件结构中，如事件数据节点的所有数据项都未使用，可显示或隐藏事件分支的事件数据节点。右键单击结构边框，然后选择**可见项>用于该分支的事件数据节点**来显示或隐藏事件数据节点。

[NI 论坛用户 *Broken_Arrow* 提供的建议。]

在循环中隐藏计数接线端

在 For 循环和 While 循环中，可选择显示或隐藏计数接线端。右键单击循环边框，然后选择**计数接线端**，可显示或隐藏计数接线端。如计数接线端已连线，则无法隐藏该接线端。

[NI 论坛用户 *somebody_that_i_used_to_k* 提供的建议。]

仅有一个输入端有连线时调换连线的位置

在早期版本的 LabVIEW 中，可以在具有两个输入且两个输入都已接线的函数上调换连线的位置。在 LabVIEW 2020 中，仅连接函数的一个输入端时，也可以调换连线的位置。单击已连线的输入端的同时按下<Ctrl>键，即可调换连线的位置。

[NI 论坛用户 *altenbach*、*tst* 提供的建议。]

设置图标上的文本

 在 LabVIEW 2020 中，可使用**快速放置**键盘快捷键<Ctrl-K>创建一个包含文本的图标。按下<Ctrl-Space>，然后按下<Ctrl-K>，LabVIEW 会将 VI 的文件名添加至 VI 图标。如果文本太长，LabVIEW 将其截断。

也可以右键单击 VI 图标，然后选择**将图标设置为 VI 名称**。

可按下<Ctrl-Space>自定义文本，输入要显示在图标上的文本，然后按下<Ctrl-K>。

[NI 论坛用户 tst 提供的建议。]

显示快速放置对话框中对象的即时帮助

 在**快速放置**对话框中选择一个对象，**即时帮助**窗口显示该对象的帮助信息。

[NI 论坛用户 elset191 提供的建议。]

连接通过快速放置插入的对象的改进

 对于通过**快速放置**在选中的连线上添加的对象，其默认连线行为有所改进。使用**快速放置**键盘快捷键<Ctrl-I>时，LabVIEW 将对象连接到最合适的接线端。下列函数的连线有改进：

- 二进制数值函数，例如：加、减、乘、除
- 二进制布尔函数，例如：与、或、异或、非与、非或、非异或、蕴含
- 二进制比较函数，例如：等于、不等于、大于、小于、大于等于、小于等于
- 复合运算
- 获取变体属性
- 字符串函数，例如：匹配模式、搜索和替换字符串、搜索/拆分字符串
- 数组函数，例如：从数组中删除、插入数组
- 商与余数
- 选择
- 插入映射表
- 插入集合

*[NI 论坛用户 D*提供的建议。]*

新增和改动的 VI 和函数

LabVIEW 2020 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

Web 服务 VI

Web 服务选板重新排列，新增了下列子选板：

- 应用程序 Web 服务器 - 包含支持部署至应用程序 Web 服务器的 Web 服务的 VI，用于配置 ESP 脚本、解密/加密数据传输、管理应用程序 Web 服务器上的 HTTP 会话。
- NI Web 服务器 - 新增了获取 NI Web 服务器认证的详细信息 VI，返回 NI Web 服务器的认证信息。

WebSockets VI

数据通信选板新增了安装 WebSockets VI 的链接，用于从 Web 用户界面以数据流方式读取和写入数据。在数据通信选板上单击**安装 WebSockets 附加软件**，通过 JKI VI Package Manager (VIPM) 安装 WebSockets 工具包。**数据通信»WebSockets** 选板上将会出现 WebSockets VI。

[特此致谢 WebSockets 附加软件的作者 Sam Sharp。]

多个错误 VI

对话框与用户界面选板新增了多个错误子选板。使用多个错误 VI 可将错误簇转换为不同格式或处理错误簇的属性。

传输层安全性函数

TCP 选板新增了 TLS 子选板。TLS 协议提供类似于传输控制协议 (TCP) 的接口，并在 TCP 之上进行加密和身份验证。LabVIEW TLS 支持重用了 TCP 协议函数。LabVIEW 支持 TLS 1.2。TLS 函数不支持 FPGA 终端。

其他新增 VI 和函数

LabVIEW 2020 中新增加了下列 VI 和函数：

- 下列 VI 可计算文件内容的消息摘要 (message digest)：
 - 高级文件选板上新增加了文件校验和 VI，该 VI 计算文件内容的消息摘要。
 - 数据操作选板上新增加了字节数组校验和 VI，该 VI 计算字节数组的消息摘要。

LabVIEW 可计算下列消息摘要 (message digest)：

- SHA-256
- SHA-224
- SHA-512
- SHA-384
- SHA-512/256
- SHA-512/224

- SHA3-224
 - SHA3-256
 - SHA3-384
 - SHA3-512
- 内存控制选板上新增了获取内存状态 VI，该 VI 返回 LabVIEW 进程使用的物理内存量。该 VI 还返回系统范围内内存使用的状态。
 - 附加字符串选板上新增了创建 NI GUID VI，该 VI 可生成一个全局唯一标识符 (GUID) 字符串。
 - 数值选板上新增了类型的范围限制 VI，该 VI 返回输入数据类型的最大值和最小值。
 - 转换选板上新增了枚举至枚举数组转换 VI，该 VI 可创建一个枚举数组，数组元素由枚举值组成。
 - 数据类型解析选板上新增了获取 LabVIEW 类（父）和成员 VI 信息 VI，该 VI 可检索变量中存储的 LabVIEW 类或接口的父 VI 和成员 VI 信息。

更改的 VI 和函数

LabVIEW 2020 中更改了下列 VI 和函数：

- 创建路径函数的**名称或相对路径**输入端的默认数据类型从字符串更改为路径。从该输入端创建的控件或常量的数据类型为路径。该输入端也接收字符串输入。
- 转换为特定的类和转换为通用的类函数新增了对接口的支持。这些函数用于将一种类或接口转换为另一种类或接口。
- 检查是否为相同类或子孙类 VI 增加了对接口的支持。除非输入数据是输入类/接口的相同类或子孙类，否则 VI 断开调用方 VI。
- 调用父方法节点更名为调用父类方法。该节点用于调用一个类方法的最近祖先实现。
- 重新设计了 TDMS 文件查看器 VI 和 **TDMS 文件查看器**对话框，现更为直观。用户可在同一页面中显示 .tdms 文件中的数据及更改显示设置，而不必在多个选项卡之间切换。
- 在之前版本的 LabVIEW 中，合并错误函数始终返回输入错误簇中的第一个错误或警告。LabVIEW 2020 支持将该函数配置为返回包含所有输入错误和警告的错误簇，方法是右键单击函数并选择**保留所有错误**。

新增和改动的属性、方法和事件

LabVIEW 2020 包括以下新增和改动的属性、方法和事件。

新增的属性、方法和事件

LabVIEW 2020 包括以下新增和改动的属性、方法和事件：

- LVClassLibrary 类（LV 类库）包含下列新增的属性和方法：
 - 父接口
 - 父库

- 为接口
- 写入父接口
- 写入父库路径
- 添加父接口
- 移除父接口
- **Application 类**（应用程序）包含下列新增的属性和方法：
 - 选板:控件名称
 - 选板:函数名称
 - **LabVIEW 类**:创建接口
- **VI 类**包含下列新增的属性和方法：
 - 执行:保存连线值
 - 控件 VI 应用更改
- **GObject 类**（图形化对象）包含下列新增的属性和方法：
 - 分组
 - 组成员引用[]
 - 已锁定
- **数组类和数组常量类**包含下列新增属性：
 - 索引矩形
- **路径类**包含下列新增的属性和方法：
 - 浏览按钮矩形
- **下拉列表类**包含下列新增的属性和方法：
 - 最后值隐藏
 - 风格
- **树形控件类**包含下列新增的属性和方法：
 - 选中项
- **多列列表框类**包含下列新增的属性和方法：
 - 选中单元格
 - 批量单元格:字体颜色
 - 批量单元格:背景色
- **项目类**包含下列新增的属性和方法：
 - 来自项 ID 的项
- **数值类、数值常量类、可重写参数接线端类**新增了配置数值控件和常量的定点行为的属性，数学函数的定点输出配置。

- 下拉列表类、枚举型类、组合框类包含下列新增事件：
 - 操作菜单激活
 - 操作菜单消除
- 控件类包含下列新增的属性和事件：
 - 快捷菜单消除

改动的方法

LabVIEW 2020 中包含下列改动的方法：

- LV 类库:包括实现 ? 方法的**考虑父类**参数更名为**考虑父类/接口**。如类从父接口继承了多个同名的方法，但不提供自身的重写方法，该方法将返回错误。
- 库:源范围:设置并传递方法包括了最新的**跳过提示 ?**参数。使用该参数指定是否在访问范围的设置应用于全部重写 VI 之前弹出提示。

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 2016 升级说明 - [upnote16zhs](#)
- LabVIEW 2017 升级说明 - [upnote17zhs](#)
- LabVIEW 2018 升级说明 - [upnote18zhs](#)
- LabVIEW 2019 升级说明 - [upnote19zh-cn](#)

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，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*。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 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。