



JETBRAINS



Reactor



Avalonia 跨平台升级、安装

Juster zhu (朱震)

Microsoft MVP | Huawei HCDE | GeneralUpdate Owner

本期内容

- 1.自动升级由来
- 2.如何解决跨平台升级
- 3.安装包制作

1.早期：物理介质与手动更新时代（1980s–1990s 初）

- 背景：软件以软盘、光盘等物理介质发行，更新依赖邮寄、官网下载或线下拷贝，用户需手动覆盖安装。
- 痛点：分发慢、成本高、用户易忽略安全补丁，版本碎片化严重，维护效率极低。
- 典型案例：DOS、早期 Windows（如 3.1）及桌面软件均无内置自动更新能力，完全依赖用户主动操作



2.萌芽：网络更新与手动触发阶段（1990s 中）

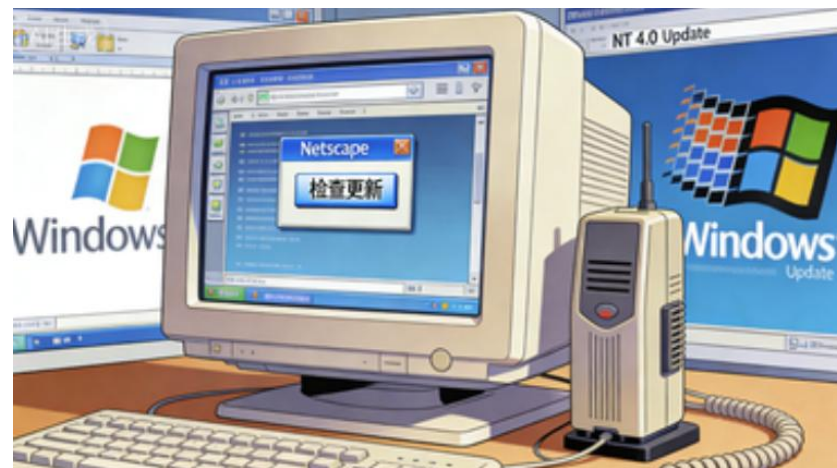
- 关键变化：互联网兴起，厂商开始提供在线更新包，但仍需用户手动检查、下载与安装。
- 代表事件：
 - 1995 年，Netscape Navigator 等早期浏览器推出“检查更新”功能，需用户主动点击触发。
 - 1996 年，Windows NT 4.0 引入“Windows Update”网页版，用户需手动访问网站获取补丁，标志着系统级更新的初步网络化。

3.突破：系统级自动更新诞生（1990s 末–2000s 初）

•里程碑：1998 年 Windows 98 集成 Windows Update 自动检测功能，成为首个支持系统级自动更新的主流操作系统，可后台检查更新并提示用户安装，无需手动访问网页。

•核心演进：

- 2000 年，Windows 2000 进一步优化更新机制，支持补丁自动下载与计划安装。
- 2001 年，Windows XP 推出“自动更新”服务，允许用户设置“自动下载并安装”，奠定现代 OS 自动更新的基础。
- 同期，杀毒软件（如 Norton、McAfee）率先实现病毒库自动更新，成为安全类软件的标配。



4.普及：跨平台与应用商店标准化（2000s 中–2010s）

•桌面端：Adobe、Apple 等厂商跟进，Adobe Creative Suite、Mac OS X（2003 年起）均加入自动更新模块，覆盖办公、设计等主流软件。

•移动端：

- 2008 年 App Store (iOS) 与 2009 年 Google Play (Android) 上线，以应用商店为中心的自动更新体系成型，用户可一键或自动更新所有应用。
- 2012 年 Google Play 正式支持应用自动更新选项，移动端更新体验标准化。

•开源生态：Linux 通过 APT、YUM 等包管理器实现系统与应用的自动更新，成为服务器领域的主流方案。

成熟：全场景智能更新（2010s 至今）

•核心特征：

- 静默更新：浏览器（Chrome、Firefox）、办公软件（Office 365）等支持后台静默更新，无需用户干预即可完成升级。
- 增量更新：采用差分算法仅下载变化部分，减少带宽消耗，提升更新速度。
- 灰度发布与分阶段推送：厂商可按用户群体分批推送更新，降低风险，如 Windows Insider、App Store 的分区域发布。
- 容器化与云原生：Docker、K8s 等技术推动服务器应用的自动滚动更新，实现零停机部署，扩展自动更新的应用边界。



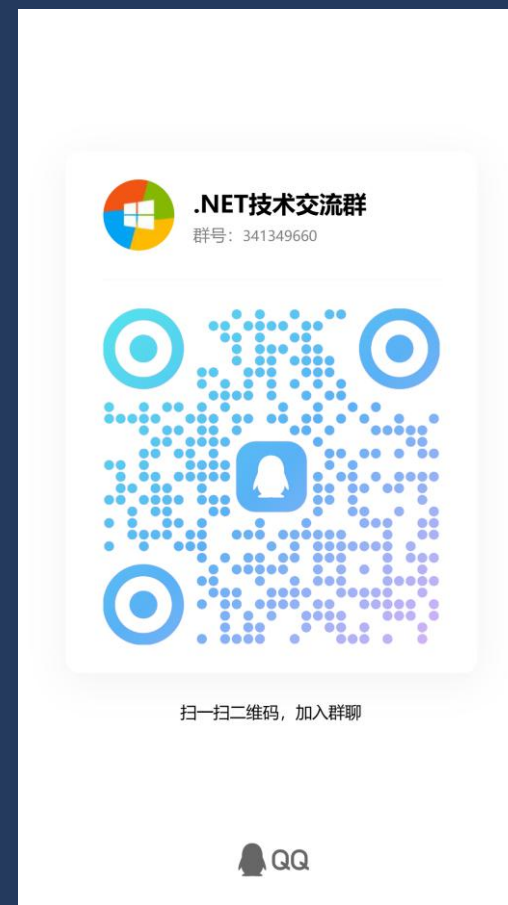
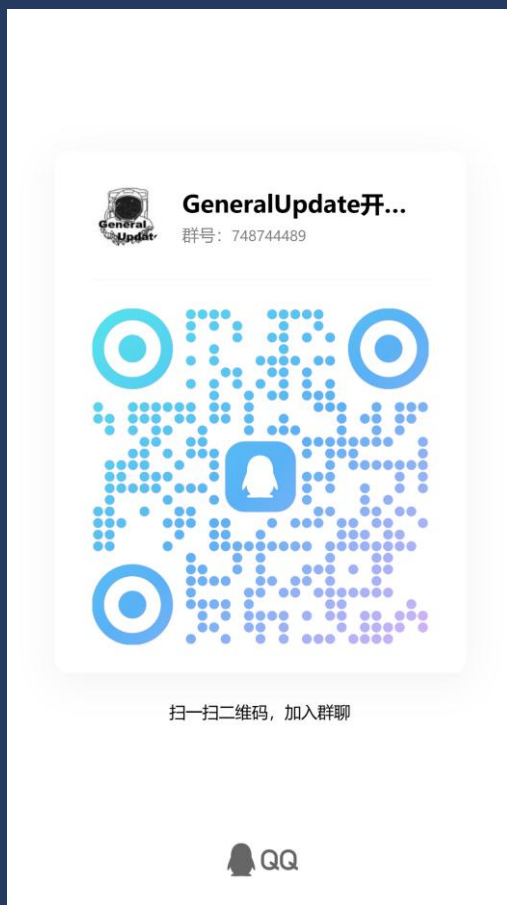




- 开源
- 商业友好
- 免费
- 公共频道

- 是什么?
- 提供哪些能力, 解决了哪些问题?
- 支持哪些框架?
-

<https://www.justerzhu.cn/docs/doc/Component%20Introduction>



详细内容

安装包制作

