

白皮书 | 2014 年 9 月

应用发布进程 自动化：构建与购买

应用交付

目录

执行摘要	3
第 1 部分：构建发布进程	4
第 2 部分：执行进程	7
第 3 部分：管理应用发布自动化	10
第 4 部分：结语	11

执行摘要

挑战

随着应用越来越复杂，对加快业务上市时间的需求持续增长，应用发布自动化已成为一项重要的 IT 功能。由敏捷开发等开发策略推动的快速开发周期，结合分布于多个不同环境的多层复杂应用程序，进一步加剧了应用部署的复杂性，并在应用生命周期方面创造了又一个瓶颈。在这一问题上耗费更多人力来消除该瓶颈并不具有成本效益。现今，我们看到越来越多的企业采用应用发布自动化来解决此问题。

不过对某些企业而言，应用发布自动化仅仅是关于给企业开发人员（或具有相关技能的其他员工）安排外壳脚本（使部署活动实现自动化）写入任务。很多时候，这一方法备受开发人员的推崇，因为他们觉得写脚本很好，“他们整天都会写代码，对于应用发布自动化，何不采用相同的办法呢？”，不过，这有时是管理层的决定；可以理解，他们对脚本的局限知之甚少（“脚本是万能的，没错，那么我们就这么干吧。”）。无论如何，必须了解对应用发布自动化基于脚本的方法的局限，才能为企业做出正确的决策。

下面我们将讨论这些局限以及 CA Release Automation（以前称为 CA LISA Release Automation）如何帮助克服这些局限。

第 1 部分：

构建发布进程

时间和努力

使用脚本构建部署进程意味着不仅必须要定义和构建进程本身，而且还必须构建基础部署功能。以前在构建脚本技术时并未考虑应用程序的部署，也就是说，在每个步骤中，不仅需考虑部署进程环境中需执行的操作，还需考虑如何用所选脚本语言实现相应操作。与此相反，何不利用已在全球各种环境得到验证的应用发布自动化平台？CA Release Automation 包含 80 多个操作包，具有逾 1400 项立即可用的“操作”，用于整合整个应用程序交付工具链中使用的解决方案和技术。根据定义，这可以大大节省时间。让脚本开发人员做他们的日常工作，而不是构建已经存在的应用发布自动化平台，会不会更有利？如果应用开发人员构建应用程序脚本，他们的时间无疑会更好地用在开发应用本身上，从而缩短产品上市时间并最终帮助企业赢得竞争，甚至发展出竞争优势。

举例说明 CA Release Automation 如何帮助节省时间，我们的一个大型财务客户希望为 20 台“日结”服务器构建一个部署进程。如果手动构建，每台服务器需花费 4 小时，总共需 80 小时才能完成部署。相比之下，使用 CA Release Automation，只需花 3 小时（比部署一台服务器的时间还短）便可完成进程构建并针对全部的 20 台服务器对进程进行测试和运行。

可重用性和最佳实践

脚本特性通常会限制可重用性和应用发布最佳实践的实施。脚本构建人员实际上可执行他们希望的任何操作。即使脚本不包含在进行进程部署或进程部署后验证前检查应用状态的步骤等基本最佳实践，也是如此。重复使用通常关于复制脚本并对特定环境副本进行更改。当必须更改脚本以适应具有不同部署进程的新版应用时，这将变得非常麻烦。

借助 CA Release Automation，所有操作集成“流”（可完成特定任务的进程片段），每个人在其进程中都可以使用该流。随着应用的发展，如果修改基础流，则所有使用该流的进程将自动使用更新后的流。CA Release Automation 以此帮助您节省时间，并促进应用发布最佳实践的使用以及各进程的可重用性。

跨服务器同步

典型的部署进程涉及针对多台服务器和多种服务器类型执行各种步骤。例如，可能必须先升级数据库服务器，然后才能升级一系列应用服务器，而使用脚本难以实现这一目标。脚本没有“服务内建概念和服务器之间的“依赖项”。两个不同的脚本如何在同步的许多不同服务器上运行呢？一种实现方法是，不在每台服务器上执行本地步骤，而是从中央服务器运行这些步骤，中央服

服务器会远程访问每台目标服务器。此方法将对这些特定步骤进行序列化，而不是在所有应用服务器上并行执行步骤。这通常会使维护窗口备受压力。写入此串行脚本后，如果部署进程改变，要更改脚本有多容易？对于具有多台数据库服务器且每台支持各自应用服务器的分布式环境，该怎么办？如何管理这些复杂的关系？

CA Release Automation 以可视化的方式创建进程，将单个操作（步骤）与相关服务器类型相对应，并在这些服务器类型之间建立起链接。此操作按逻辑完成，无需指定该进程涉及的特定服务器，因此可根据需要快速更改服务器及其依赖项，而不必修改进程。

图 A.

展示跨服务器同步的多层进程

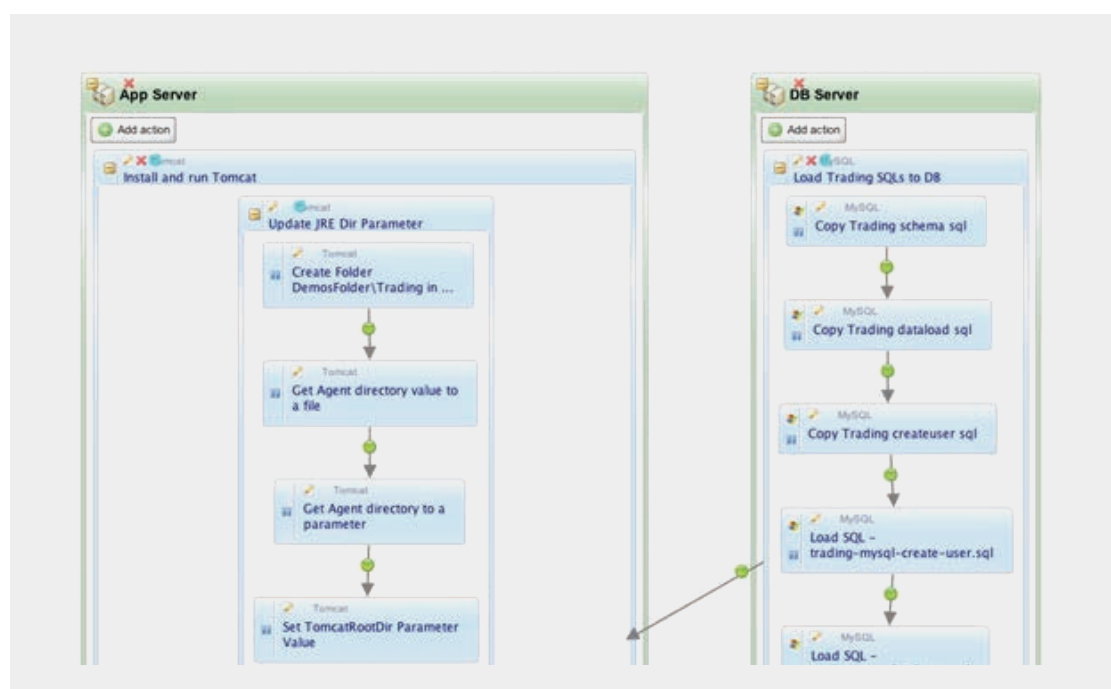
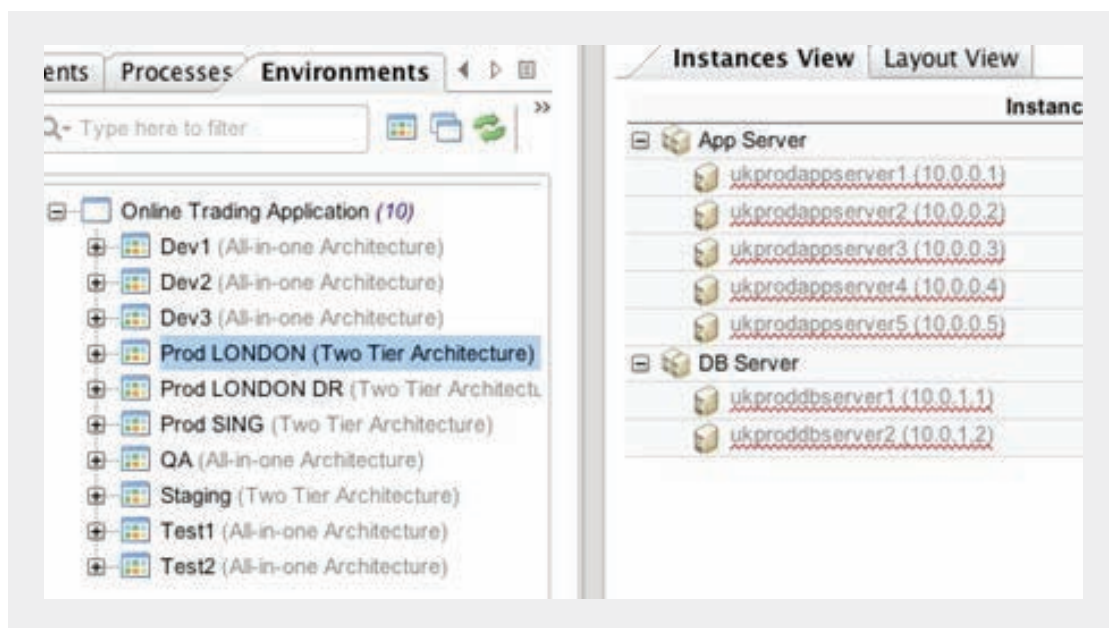


图 B.

环境配置



该环境中的一系列服务器及其相互关系得到良好维护（例如，哪个数据库服务器与哪个应用程序服务器一起工作）。

立即可用的集成

以前在构建脚本语言时并未考虑应用发布的自动化。因此，它们只包含复制文件、运行外部命令等基本步骤的功能。

CA Release Automation 拥有 1400 多项立即可用的操作类型。这些类型的操作包括：基础文件和文件夹管理、字符串/参数操控、程序包安装（如 .exe、.msi 和 rpm 的安装）、读取/写入 XML 和 INI 文件、命令行执行、SOAP 调用等。此外，还有用于各种技术的特定操作，例如操控 VM 环境，使用 EC2、F5 Big IP、WebSphere、Weblogic、数据库等，以及用于与应用程序生命周期中的其他工具（如 Subversion 和 TFS）集成的操作。

多个环境

如果使用脚本，通常需查找内嵌于这些脚本中的服务器名称（通常在多个地方）。如果脚本稍后用于不同的环境，必须以不同的服务器名称创建新版脚本。这给未来创建和维护这些脚本都增加了额外的时间。

另一方面，CA Release Automation 有意将逻辑进程从将运行该进程的实际服务器中分离。该进程在最后一刻（即运行时）才与环境信息建立“连接”。因此，决定在不同的环境中运行进程就是一个简单的任务，即在用户界面中定义该环境并指引相应进程在该环境中运行。无需退回去修改实际进程以适应新的服务器名称。除了节省时间外，此平台还可通过预生产环境使进程变得“成熟”，以便将同一进程放心部署到生产中。

第 2 部分：

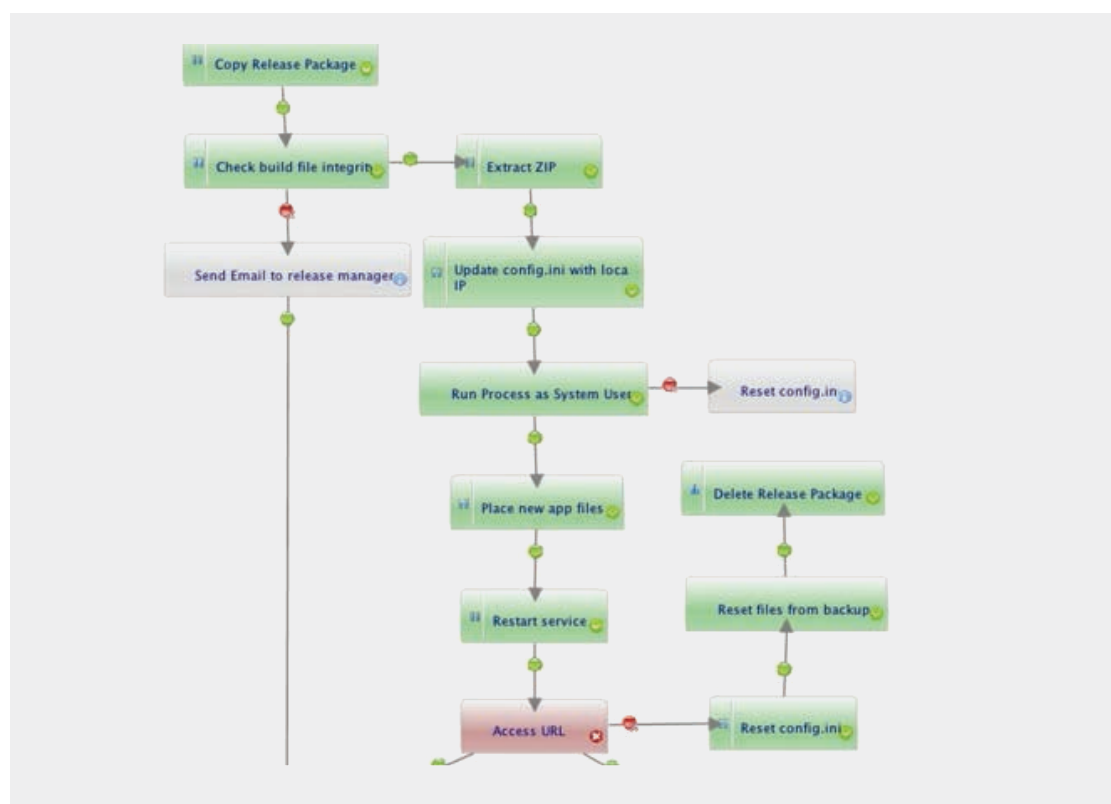
执行进程

多个环境

在执行部署进程的过程中，务必了解一路上在每个步骤会发生什么。这需要清楚地了解哪些步骤起作用，最重要的是，哪些步骤没有起作用。脚本是一个黑匣子。万一发生故障，确定哪个步骤失败等这样的简单事情并不容易，因此速度也不会很快。查找这一问题便会花掉大把时间。即使从脚本中获得结果，通常也是用“技术性”语言而不是人类语言所写。

借助 CA Release Automation，执行过程中会显示进程，每个操作在执行时会改变颜色，清楚显示哪些步骤通过（绿色），哪些失败（红色）。每个操作都有一个人类可读的名称，如“复制发布包”，因此发生故障时可立即清楚了解进程中的哪个步骤失败了。

图 C.
进程执行



故障可立即清楚识别，并可通过两种方式来处理，一是构建“故障”路径（如上述示例），二是将进程暂停在该点并将该问题通知相关人员，以便他们解决问题并继续后续步骤。

暂停/恢复

运行脚本时，它会完成自己的协议，或在未完成的情况下在随机的地方被中止（取决于操作人员何时按 CTRL-C）。中止后，脚本必须再次从头开始。

借助 CA Release Automation，可清楚查看执行期间进程所处的阶段，且操作人员可随时按“暂停”按钮。如上所述，失败的步骤也可自动触发暂停。暂停后，可恢复进程，操作人员可决定是否应再次执行当前步骤，或是否应跳过该步骤。无需从头开始重新启动进程。

执行时间

通常，写入脚本是为了处理其中一个应用。然后在部署中，操作人员需以给定顺序运行一系列脚本，并等待每个脚本完成运行。这是一种非常低效的方法。CA Release Automation 提供一个用于执行整个进程的“运行”按钮，可在多台服务器上执行多个步骤，并如前文所述使这些步骤实现同步。

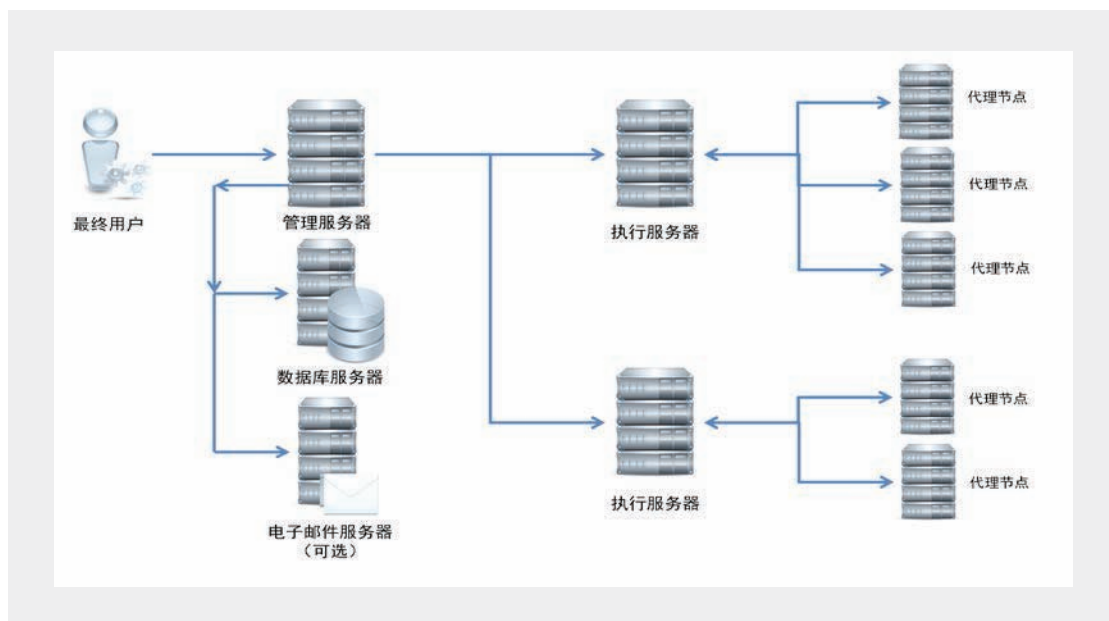
文件分发

在传统的脚本方法中，如果进程要复制 100 台应用程序服务器，则该脚本文件需从源位置复制 100 次到每个目标应用程序服务器。

使用 CA Release Automation 架构，只需从源位置执行一次到最近的“执行服务器”的复制操作，然后将文件一次性传输到另一个执行服务器，并在本地将文件复制到每个目标应用程序服务器。通过将执行服务器放置在正确位置，可降低网络带宽利用率。这对于需要克服受限带宽连接以进行复制的情况特别有用。

图 D.

CA Release
Automation 架构。



报告

能够执行部署是一方面，能够获取所有结果，提供报告信息并证明合规性便更加困难了。基于脚本进行应用部署并不能满足报告要求。

另一方面，CA Release Automation 将所有信息存储在一个中央存储库中。这包括每次执行的信息，例如：谁运行进程，运行的是什么进程以及在哪个环境中运行，进程运行了多久，这提供了该运行的每个步骤的详细信息，以及时间戳和状态信息，还有服务器信息。这些信息可在详细的报告中找到，也可在更“便于管理”的仪表板中查找。

通过这些信息，所有利益相关者都可了解应用环境中所发生的状况，并根据需要将这些信息用于与合规性相关的目的。

队列管理

执行部署时，如果另一个进程同时在这些相同的服务器上运行，即使不是灾难性的，也可能带来破坏性的后果。如果一个进程尝试使用其复制到相应服务器的文件，但同时另一个脚本用不同版本的文件替换了该文件，会发生什么情况？若使用脚本，则无法停止同时运行冲突的脚本。

CA Release Automation 提供内置队列管理功能，用于阻止需要相同资源的进程同时运行。需要使用已经在运行应用程序发布进程的服务器的进程会自动排队，直到前一个进程完成。这样可在每次执行的持续时间内维持该执行的环境。

谁运行进程？

为了解脚本失败的确切原因，通常必需涉及到脚本开发人员。不熟悉脚本或脚本语言的人会觉得难以查找问题的根本原因，因而耗费大量时间。CA Release Automation 提供关于进程状况的详细信息，这有助于用户以易于理解的方式清楚了解失败的步骤（如果有）。因此，在每个环境的部署进程中不需要开发人员来“解释”结果。

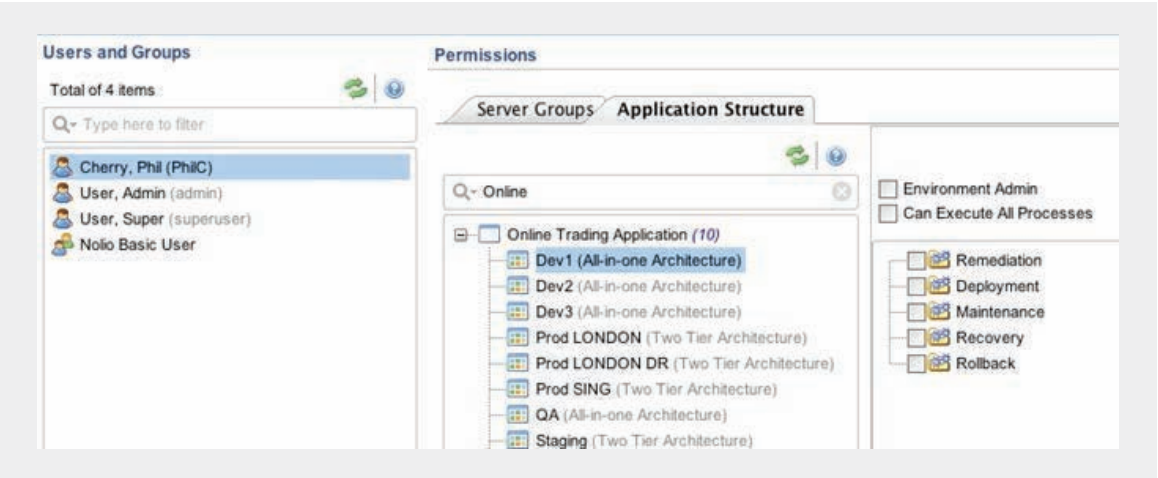
第 3 部分：
管理应用发布自动化

访问控制

构建脚本后，如何控制该脚本的使用？更重要的是，如何按用户、应用、环境和进程控制受限访问？例如，开发团队可以使用这些脚本将应用程序部署到他们的开发环境中，但是对于过渡和生产环境，该进程通常由其他人控制。如果依赖脚本的操作系统级别访问权限，这将难以控制，而且也很容易忘记谁具有每个环境中的每个脚本的权限。

借助 CA Release Automation，可为每个用户设置精细的权限，限制他们访问特定应用程序、环境和进程。因此可轻松控制操作人员、操作内容和位置。在规定一个人不能同时部署到预生产环境和生产环境的环境中（通常称为“分离控制”），这尤为重要。

图 E.
权限管理



进程生命周期

脚本一旦构建完成后，便几乎无法停止其运行。借助 CA Release Automation，我们通过“发布”功能将设计与执行分开。只有特定人员才能发布进程，因此这里引入了先“批准”进程再发布并用于环境中的概念。

协作平台

继前述功能，CA Release Automation 提供了使用脚本无法实现的协作平台。不同的人员可在整个进程生命周期中共同承担责任，并限制参与该生命周期的相应部分。例如，开发人员可以构建个别步骤（操作）和流（如前所述），同时进程专家将该进程集合在一起，发布经理批准并“发布”进程，而环境负责人执行实际的进程。此分离控制并非强制性的；管理员可根据需要定义角色和权限。这意味着，不是只有能够构建或理解脚本语言的人员才可参与应用程序部署。

锁定环境

操作人员需要有相关用户名/密码才能在目标服务器上运行相应脚本。借助 CA Release Automation，每台服务器的凭据可保持私密，用户登录 Release Automation，他们的特定权限使他们能够/限制他们仅能够执行允许的操作。系统会跟踪执行的所有部署（含时间戳、与该执行关联的用户等）。通过不为用户提供环境的直接访问权限，可最小化或者甚至消除配置漂移，并且可大大降低服务级别受新应用部署影响的风险。

第 4 部分：

结语

应用发布进程自动化的概念获得认可后，接下来的问题便是解决如何实现所需的自动化。在此阶段，通常会建议使用内部技术人员并“利用脚本的强大力量”来完成此任务。表面看来，这似乎是个明智的方法。内部通常有精通脚本语言的技术人员，那么为何还要花钱开发新产品并针对该产品的使用进行培训。

仔细审视自动化这一理念非常重要。

出于某种原因，脚本通常被视为“安全的选择”，仅仅因为它是已知的实体。但可以肯定的是，安全的选择不会依赖于专为手头上的任务开发的产品。

任何自动化解决方案都需要获得企业的广泛认可，而不仅仅是在知道如何写脚本的特定技术人员的范围内。解决方案还需要具有多样的功能，提高可重用性，促进最佳实践，并支持复杂的跨服务器依赖项和同步，同时允许用户轻松了解每个阶段的发布状态。可行的解决方案还需要使非开发人员能够参与应用部署活动，有把握地管理每个环境中的操作人员和操作内容。此外还需要支持整个企业所有类型的利益相关者的报告需求。

不仅如此，该解决方案需要在各种环境中得到验证，随着要求的变化和技术平台的发展（它们始终在变化），所选发布自动化平台能够支持这些新要求，同时提供在应用生命周期内与其他工具集成的能力。

应用发布自动化解决方案可实现上述所有目的，无需占用企业的宝贵资源即可构建并维护该自动化功能。更好地将时间利用在提供企业获得成功所需的创新解决方案上。

内部开发的脚本做不到，而 CA Release Automation 能做到。

CA Release Automation

CA Technologies 是一家应用发布自动化解决方案提供商，旨在针对发布和管理分布式、多层数据中心应用的生产必需的应用发布部署和其他操作服务任务实现自动化。该公司的解决方案让IT 运营团队能够在各种应用层和环境中的应用程序发布工作流建模，为应用的更改、维护、故障排除和回滚实现自动部署。工作流在异构数据中心基础结构中执行，提供准确、灵活的模型，用于管理物理、虚拟和云服务器中的应用更改。CA Technologies 针对数据中心自动化采用以应用为中心的方法，这巩固了应用孤岛之间的应用服务知识，包括开发、QA和运营，对应用操作进行集中化控制。



联系 CA Technologies，网址：ca.com/cn



CA Technologies (NASDAQ:CA) 开发促进企业转型的软件，为他们抢占应用经济的先机。软件是各行各业的核心。从规划到开发再到管理和安全性，CA 正与全球各地的公司开展跨移动、私有和公共云、分布式和大型机环境的合作，以改变我们的生活、交易和沟通方式。要了解详细信息，请访问 ca.com/cn。