
白皮书

Solutions Business Manager 企业生产路径

目录	页码
Solutions Business Manager 开发流程简介.....	1
生产路径参与者	1
标准安装拓扑	2
使用沙箱开发系统	4
在生产中修补流程应用程序	6
总结	10

使用熟悉的拖放界面可在 Business Manager Composer 中直观设计工作流程、表单和其他流程项目。

Solutions Business Manager 开发流程简介

OpenText™ Solutions Business Manager 旨在简化业务流程的开发和部署。使用熟悉的拖放界面可在 Solutions Business Manager Composer 中直观地设计工作流程、表单和其他流程项目。流程应用程序处于可试用的阶段时，您可以单击某个按钮在测试环境中进行验证和部署。您可以使用环境中的 Application Administrator 在模拟环境下设置用户、组和项目。一切准备妥当之后，Application Repository 可以轻松将配置的流程从模拟环境升级到生产环境。本白皮书讨论标准生产路径和替代沙箱方法。

生产路径参与者

生产路径介绍如何将面向最终用户的设计部署到生产环境中。在 Solutions Business Manager 中，涉及 Solutions Business Manager Composer、Solutions Business Manager Application Repository 和各种 Solutions Business Manager 运行时环境之间的相互影响。这些组件将在下面介绍：

Solutions Business Manager Composer

Solutions Business Manager Composer 是用于设计流程应用程序的客户端应用程序。在 Composer 中，可定义用来表示项目的收集数据，如何在组织内移动和更新该项目，组织内的角色，以及用户用于查看和修改项目数据的表单。

Solutions Business Manager Application Repository

Solutions Business Manager Application Repository 具有两种用途。首先它可以充当单个设计组件（如表单或工作流程）的源控制储存库。在本白皮书中，另一个更重要的用途是此应用程序可以将流程应用程序部署到多种环境以及在不同环境之间移动流程应用程序。¹ Application Repository 将发布到其中的流程应用程序的每个唯一版本储存起来，并且保留与运行时环境之间的所有重要交互记录。它是 Solutions Business Manager 生产路径的核心参与者。

¹ 虽然开发人员可以直接通过 Business Manager Composer 将流程应用程序从内部部署到某一环境中，但是通常先将流程应用程序发布到 Application Repository，然后再部署到目标环境。

环境

生产路径的另一个参与者是环境。环境是指面向管理员和最终用户的整套运行时服务。这些服务包括应用程序引擎（向最终用户呈现在 Composer 中设计的表单）、编制引擎（执行用于调用 Web 服务的自动化 BPEL 流程）、通知服务器（将通知分派给感兴趣的用户）以及邮件客户端（处理收到的电子邮件以及其他运行时服务）。

Solutions Business Manager 的安装通常带有多种环境。开发环境用于测试变更，而不影响最终用户。模拟环境用于复制生产中使用的配置，以及测试所配置的流程应用程序。生产环境让您创建的流程应用程序和解决方案为 Solutions Business Manager 业务用户创造价值。

每种环境还包含 *Solutions Business Manager Application Administrator*，支持管理员配置和优化流程应用程序的运行时方面。通常而言，您可使用 Composer 创建运行系统在不同环境中的通用部分（设计概念），而使用 Application Administrator 配置运行系统适合特定环境的部分。例如，用户、组、项目和通知都是在 Application Administrator 中配置的管理概念。它还用于“覆盖”原设计，以允许流程应用程序随着项目而变化。

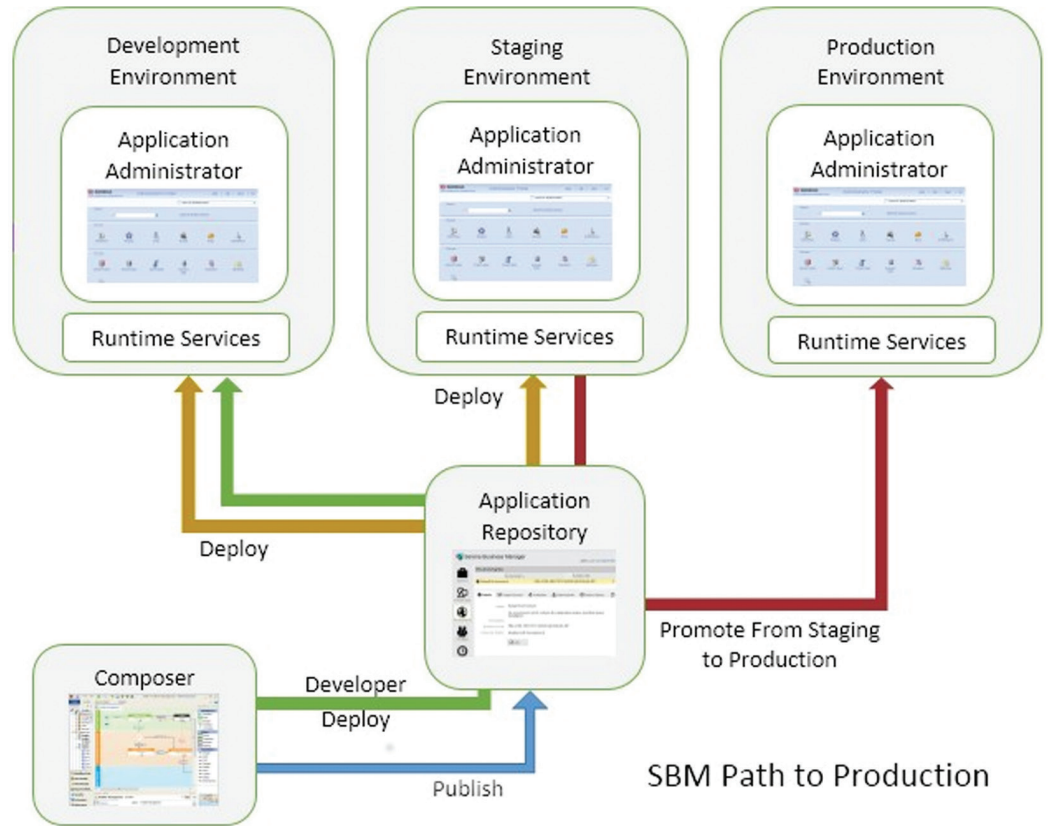
Solutions Business Manager 的安装通常带有多种环境。

标准安装拓扑

Solutions Business Manager 可以使用各种拓扑安装，足以满足我们客户的各种需要。标准安装具有三种环境，分别是开发、模拟和生产环境，并且使用单个 Application Repository 来协调流程应用程序部署到不同环境以及在环境之间升级流程应用程序。

下图说明了这种安装类型中 Composer、Application Repository 和运行时环境之间的关系。

开发期间,业务开发人员首先创建或修改一项流程,继而优化流程应用程序,将其部署到开发环境并进行测试,周而复始直至准备好将设计投入模拟环境。



2 Application Repository 中的环境具有可设置为“启用部署”的 Composer 属性,以便允许从 Composer 直接部署。如果将该属性设置为“启用开发部署”,Composer 用户可以选择这样一种部署:无需归还更改,或者无需在 Application Repository 中创建所部署蓝图的版本。

生产路径生命周期

以下步骤介绍标准安装的典型生产路径生命周期。

业务开发人员首先使用 Composer 创建或修改流程应用程序。开发期间,开发人员可优化流程应用程序,将其部署到开发环境并进行测试,周而复始直至准备好将设计投入模拟环境。为促进开发工作,管理员可以选择关闭这些开发部署的版本控制。²

设计完成后,开发人员将流程应用程序发布到 Application Repository,管理员将在此处将其部署到模拟环境。

在模拟环境中, 将使用 Application Administrator 和报告界面来配置流程应用程序, 以提供生产所需的运行时功能和管理功能。例如, 定义用户并授予相应特权和角色。还可以定义项目, 并对项目实施设计覆盖。完全配置流程应用程序后, 将在此环境中执行验收测试, 以确保使其做好准备, 可供用户投入生产环境。

最后, 流程应用程序通过生产前的验收之后, 会通过 Application Repository 升级功能从模拟环境升级到生产环境。升级过程不同于部署过程, 因为部署过程仅传送在 Solutions Business Manager Composer 中创建的设计, 而升级过程除了传送底层设计外, 还会将运行时和管理设置传送到目标环境。

由于开发过程因公司而异, 您使用的 Solutions Business Manager 配置需要相应调整。

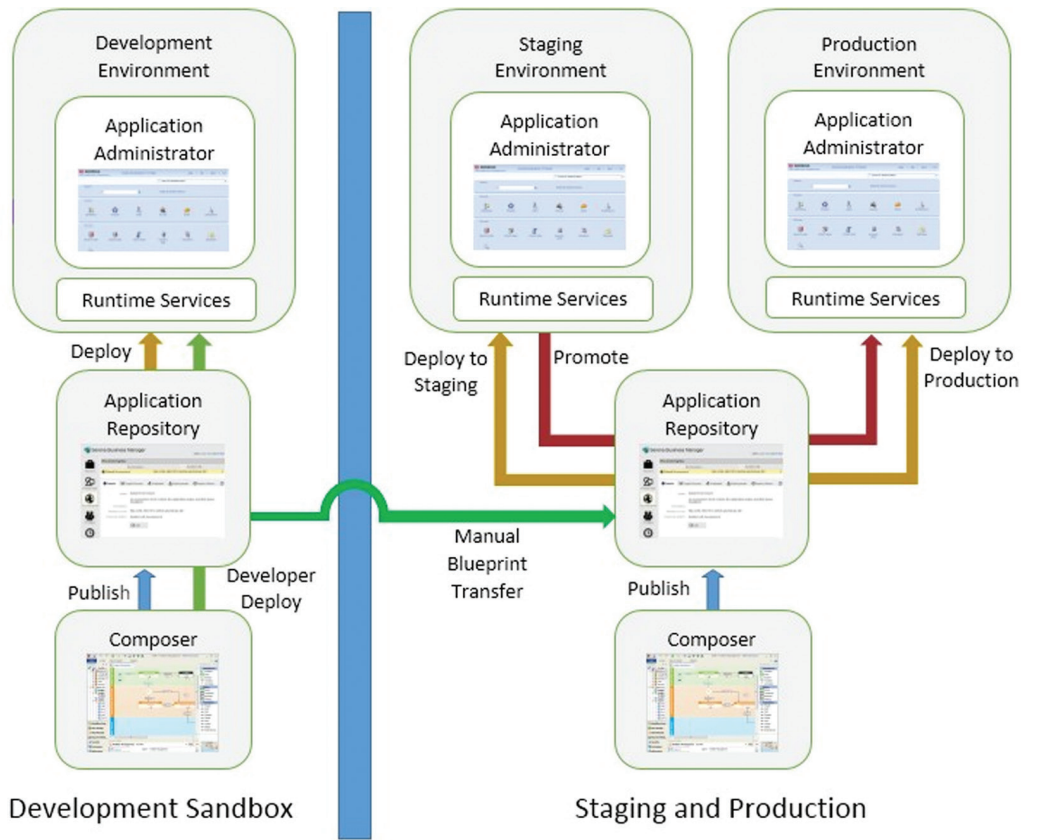
使用沙箱开发系统

由于开发过程因公司而异, 您使用的 Solutions Business Manager 配置需要相应调整。在本部分中, 我们将针对必须以物理方式分隔沙箱开发系统与模拟和生产系统的环境, 介绍其适用的生产路径。这种配置类型的优势包括:

- 简化开发和生产之间的用户和特权模型的管理
- 物理分隔让生产环境实现物理安全性
- 支持断续式开发工作

下图说明了沙箱系统中 Composer、Application Repository 和运行时环境之间的关系。

可以授予开发人员完全管理权限,以便在 Application Repository 和开发环境中执行任何需要的操作,而没有意外更改生产系统的风险。事实上,开发人员甚至不需要是模拟/生产系统中的用户。



拥有单独的开发沙箱系统时,该系统的用户无法访问模拟或生产环境。可以授予开发人员完全管理权限,以便在 Application Repository 和开发环境中执行任何需要的操作,而没有意外更改生产系统的风险。事实上,开发人员甚至不需要是模拟或生产系统中的用户。

利用此拓扑,流程应用程序开发人员完全在开发沙箱中工作。完成设计并准备好投入模拟环境之后,流程应用程序将从 Composer 或 Application Repository 导出为蓝图 (.msd) 文件,然后导入模拟和生产系统上的 Application Repository 中。这种等效的手动方法能将流程应用程序从沙箱系统发布到模拟和生产系统中的 Application Repository。

您还可以对沙箱中的流程应用程序进行管理更改, 然后使用 Application Repository 的手动升级功能将这些更改传送到模拟系统。在开发环境中进行管理更改后, 可以从中获取快照, 将其从沙箱 Application Repository 导出为快照 (.mss) 文件, 将其导入模拟和生产系统中的 Application Repository, 然后在此完成升级。

管理员接着将流程应用程序部署到模拟环境, 而模拟环境使用 Application Administrator 和报告界面按前文所述继续操作。

最后, 配置好流程应用程序并准备好投入生产后, 管理员可使用 Application Repository 将配置的流程应用程序升级到生产系统。

配置好流程应用程序并准备好投入生产后, 管理员可使用 Application Repository 将配置的流程应用程序升级到生产系统。

在生产中修补流程应用程序

在上面概述的流程中, 流程应用程序的设计决不能在模拟和生产系统中修改。这意味着, 设计的“可靠来源”始终是开发系统及其连接的 Application Repository。这使得开发流程变得简单, 但在实践中并不是始终可行。考虑替代方法时, 请考虑以下问题:

- 是否要修改当前用于生产的流程应用程序的版本?
- 是否仅将目标更改应用于生产环境?
- 是否已将相同更改应用于流程应用程序的现有开发版本?
- 对于要添加到生产环境且具有 ID 的项目, 是否确保已将相同项目 (具有相同 ID) 添加到流程应用程序的开发版本?

本部分讨论直接更改模拟或生产系统时面临的挑战, 并且介绍应对这些挑战的方法。

要直接更改生产流程应用程序，需要获取已部署到生产的流程应用程序版本的访问权限，然后更改其设计并重新部署到生产环境。

在生产中打补丁的要求

要直接更改生产流程应用程序，需要获取已部署到生产的流程应用程序版本的访问权限，然后更改其设计并重新部署到生产环境。如果开发工作尚未完成，需要确保其中的任何更改不会意外投入生产中，而仅是您需要的补丁更改投入生产中。您还需要确保将生产更改合并返回到现有开发系统，以防在后续部署当前开发的更改时将其遗漏。

如果生产更改会引入状态等项目（项目在系统中通过唯一标识符而非名称进行识别），修补系统时可能会产生问题。如果在开发系统中独立添加具有相同名称的状态，它将不会与您以前添加到生产的状态具有相同的标识符。这意味着，后续将该流程应用程序部署到生产时，将具有两个名称相同的状态，其中一个不再参与工作流程。此问题的解决方法对于标准安装和含沙箱开发系统的安装有所不同。

标准安装中的打补丁方法

使用单个 Application Repository（即未使用沙箱系统）时，可以使用 Composer 补丁环境功能更改已部署到生产环境的设计。在 Composer 的补丁环境中打开流程应用程序时，使用的是流程应用程序的带标签版本。您可以借出设计元素，修改设计元素，归还这些元素，然后发布和部署流程应用程序的修改版本。这样做可以最大限度地减少流程应用程序现有新开发的中断。

您在补丁环境中可更改的内容具有一些限制，以防后续出现 ID 冲突。例如，您无法直接添加新状态或字段；但是，您可以根据开发提示复制状态或字段。这将防止生产系统和这些项目的开发提示之间出现分歧。因此，如果要添加字段到补丁生产环境，需要先在开发系统中添加该字段，再在补丁环境中借出表格，然后使用表格的“添加现有字段...”上下文菜单项将其复制到补丁环境中。使用所需更改更新补丁环境后，可以发布流程应用程序的打补丁版本，然后将其部署到模拟或生产环境。

打开补丁环境

通过选择“文件”菜单中的“打开...”，调出“打开流程应用程序”对话框。选择“查找范围: 储存库”单选按钮, 然后选择要修补的流程应用程序。单击“打开带标签版本...”按钮。将显示一个对话框, 其中包含已发布版本的列表。选择生产中的版本并单击“确定”。如果以前未打开此版本作为补丁环境, 将显示一个对话框, 以便根据选定的版本标签创建补丁环境。单击“是”创建补丁环境。在补丁环境中使用 Composer 时, 将在流程应用程序名称和版本标签后面的标题栏中显示 “[补丁]”。

标准安装中的打补丁方法

修补与开发系统隔离的生产环境时, 同样会出现上面概述的问题。但是, 由于系统相互隔离, Solutions Business Manager 无法防止您产生 ID 冲突, 因此需要采取措施确保所有生产更改与当前开发工作的后续部署保持一致。您可以采用多种方法来解决这一问题。

使用补丁环境

如果您在沙箱 Application Repository 中具有当前部署到生产的流程应用程序版本, 可以根据其标签在沙箱 Composer 中创建补丁环境, 按照上述方式进行更改, 然后将设计导出为蓝图。此时可以将蓝图导入生产系统中的 Application Repository, 然后部署到模拟或生产环境。如前文所述, 必须使用在补丁环境中直接进行的任何更改来更新开发提示。

使用 Composer 比较/合并功能

要使用 Composer 合并功能确保一个系统中所做的更改准确反映在另一个系统中, 有两种方法。您可以将更改从沙箱系统复制到生产系统中, 反之亦然。采用哪种方法取决于要在生产中进行的更改是否已在沙箱系统中做了更改。

将更改从沙箱合并到生产中。如果已在沙箱系统中进行生产更改, 您会希望该更改合并到生产系统中。为此, 先从沙箱系统获取流程应用程序的蓝图 (.msd) 文件, 方法是使用“文件 -> 导出...”菜单项。接着打开连接到生产 Application Repository 的 Composer 实例, 然后打开生产中的流程应用程序版本。如果最初导入到生产 Composer 并经过发布后进行部署, 它将出现在带有蓝色和金色图标的“打开流程应用程序”对话框中 (“储存库”选项)。如果最初导入到 Application Repository 并从中部署, 它将显示在带有红色图标的对话框中。

Solutions Business Manager 无法防止您产生 ID 冲突, 因此需要采取措施确保所有生产更改与当前开发工作的后续部署保持一致。

如果要在两个状态之间添加过渡状态,可以在生产系统和沙箱系统上都独立添加该状态。

在 Composer 中打开流程应用程序的生产版本后,可以使用“文件->比较->与本地文件...”菜单命令,将包含流程应用程序沙箱版本的蓝图与生产版本进行比较。您接着可以将所比较应用程序(沙箱系统中的蓝图)中的“复制以打开流程应用程序”上下文菜单项,用于要在生产环境中复制的任何项目。此方法将保留设计元素的唯一 ID,因此以后无需担心 ID 冲突。

将更改从生产合并到沙箱。您可以先在生产系统中进行更改,然后将这些更改重新合并到沙箱系统,方法是反向使用上面概述的方法。在这种情况下,将从生产系统获取流程应用程序的蓝图文件(方法是从 Composer 导出),在 Composer 中打开流程应用程序的沙箱版本,然后合并到开发系统。

并行进行次要更改

对于未创建具有唯一运行时 ID 的项目的更改,可以在系统之间并行进行更改。例如,如果要在两个状态之间添加过渡状态,可以在生产系统和沙箱系统上都独立添加该状态。但是,必须确保未将以下任何设计元素独立添加到系统:

- 表
- 字段
- 选择
- 应用程序工作流程
- 状态
- 报告定义
- 角色
- 应用程序
- 编制

否则当沙箱系统中的流程应用程序版本最终部署到生产时, 将导致冲突。您可选择使用上述“比较”功能, 确保在两个系统中都准确进行此类次要更改。

总结

标准安装的 Solutions Business Manager 在所有环境之间共享一个 Application Repository 实例, 拥有简单且受控的生产路径功能。如果使用沙箱系统用于开发工作, 可能会增加不同设计之间出现分歧的概率。在这种情况下, 需要注意可能会产生问题以及本文中介绍的合并更改的方法。³

如需了解更多信息, 请访问

www.microfocus.com/opentext

³ 另见《Solutions Business Manager 开发最佳实践》, 了解如何将合并更改难度降至最低。

与我们联系

[OpenText 首席执行官 Mark Barrenechea 的博客](#)

