CA Release Automation 5.0

Migration best practice guide

Author : Puzey, Keith

Version: 1.2.1

Filename: CA Release Automation 5.x Migration Guide.docx

Date: Tuesday, 15 July 2014

Introduction

CA Release Automation 5.0 introduces several new concepts and the intention of this document is to explain this new functionality and how applications can be modified to utilise these exciting new features.

In place upgrades are supported form CA Lisa Release Automation 4.5.1 or higher and any existing applications will be upgraded to 5.0 and require no modification but for the application to utilise the new features in CA Release Automation 5.0 the upgraded application needs to be modified. This document is the guide to the various changes required.

The order for upgrading CA Lisa Release Automation is as follows:

1) NAC / Data Manager
2) All Execution Server
3) All Agents
4) Upgrade any CI server Plugins that are being used
5) Modify the REST calls being used by any external systems.

Note: Release Automation 5.0 requires additional network ports as listed below:

1) TCP Port 61616 between NAC and all Execution servers.
2) TCP Port 8083 between the Web browser (ROC) and NAC connection, this port is required when using the Action pack manager screen.

Release Automation 5.0.x Marquee Features

CA Release Automation 5.0.x includes many new features and updates the following list are the primary Marquee features:

- Improved Release Visibility including:
  - Deployment Pipeline Report
  - Deployment Comparison Report
- Artifact Caching and Deployment
- Centralised Action Pack Management

Release Automation 5.0 Concepts

Applications

An application is a design element that represents an automated configuration and deployment for a server architecture. To create a deployment for a server architecture, the user permissions, processes, and server types must be associated with the same application.

Processes

A process is a repeatable workflow of actions that perform operations in a server architecture. The main use case for a process is to deploy one or more software components to a correctly configured environment.

Actions

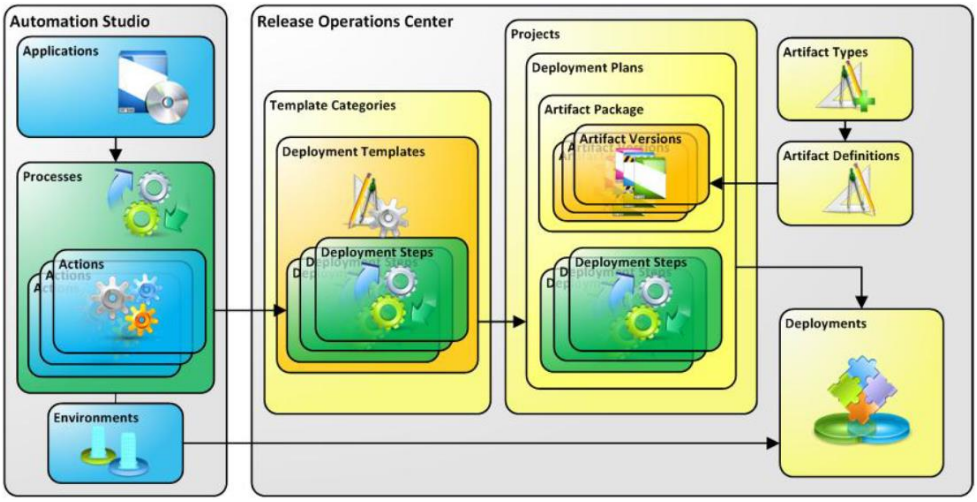
An action is an individual operation to perform in the target environment. Organize actions in categories, flows, and loops to create commonly used workflow segments.

Environments

Create reusable environment configurations to assign to processes and deployments.

Release Operations

Release Operations Center enables you to create reusable deployment templates, deployment plans, and artifact packages. A release is a combination of the deployment plan with an environment configuration to create a deployment.



Template Categories and Deployment Templates

A template category is a container element that associates a set of deployment templates with an application. Deployment templates that share common steps or technologies should be grouped together within the same template category. Each deployment template represents a particular workflow of deployment steps that are required to deploy the application where each step is a published process created in Automation Studio.

Artifact Types, Definitions, Versions, and Artifact Packages

An Artifact type is aligned with the components within the application and includes attributes specific to the artifact type as well as default attributes

Artifact definitions are aligned with server types and contain values for the attributes defined within the artifact type.

Artifact objects enable you to build a reusable library of software component to deploy artifacts.

Projects, Deployment Plans, and Deployments

A project specifies an application version, typical examples of projects could be an internal project name or for customers using agile the sprint name would be used to name the project. A deployment plan is created by using a deployment template and selecting the artifact packages required for a deployment, a typical name for a deployment plan is “nightly build *ddmmyyy”*. The final step assigns an environment to create a deployment.

XML Manifest

CA Release Automation allows you to generate XML manifest files that you can use to specify release parameters. CA Release Automation generates an XML template that includes tags to identify all the relevant information

Migration of 4.x applications to 5.0.x

During an in place upgrade or release automation 4.x any internal parameters that have an environment value will be automatically changed to a new parameter scope RA 5.0 “Environment” parameter.

A Release Automation 5.x deployment template consists of three primary components

1 Initialization Steps

The Initialization step in the CA Lisa Release Automation 4.x has been enhanced within RA 5.0 and is now in two parts as described below, when reengineering a 4.x application any INIT steps need to be reviewed to identify if the functionality should run as part of the Pre-Plan or Pre-Deployment:

a) The Pre-Plan step is used to add an artifact package to your deployment plan, the artifact package comprises of all artifacts required for the deployment. The Pre-Plan step can also be used to update release and template parameters.
The pre-plan step runs after deployment plan creation.
b) The Pre-Deployment step is used to update your deployment according to the selected environment. When the deployment plan is created the target environment is selected and this means that the Pre-Deployment step can be used to set parameters based on the target environment. The Pre-Deployment step runs every time a deployment is created.

2 Deployment Steps

When reengineering 4.x applications the core components within the application should be broken down into functional modules and published as individual processes. These processes are bought together as deployment steps and can be ordered using the dependencies functionality within the deployment plan,

3 Post Deployment Steps

Post Deployment steps would normally include any testing or validation activities required to ensure that a deployment has been successful.

Best Practice for Reengineering 4.x applications

Large processes should be reviewed and core functionality should be broken into smaller processes that can be reused.

When reviewing existing applications flows that previously used xpath queries to parse xml files and set parameters can be replaced with Collection Elements, this new type of flow loop allows for a much more efficient method for handling this functionality.

Two new Parameter scopes have been added with 5.x and when reengineering a release automation application the parameters already in use should be reviewed to understand if one of these new parameter types should be used, the two new parameter scopes are:

a) Environment Parameter – The value of an environment parameter can either be set in the ROC / administration / Environment Configuration or via an action. When creating a deployment any environment parameters within the application will be updated with the relevant environment parameter.
b) Release Parameter – Release Parameters can be updated using the manifest file that can be created from the deployment template or using an action.

The Artifact deployment methodology within Release Automation 5.0 has been enhanced so there is no longer the requirement to specify how an artifact is retrieved. The artifact defintions are exposed within the Designer UI as parameters, if these parameters are used within a published process that is used within a deployment plan the artifacts are automatically either copied to the relevant execution servers that will take part in the deployment or to an artifcat cache on the agent machines. Any artifact copy processes within the application should be reviewed and modified to utlise this new functionality. Utilising the artifact cache will speed up the deployment time as the artifacts are copied locally instead of over the network.

The use of User Input parameters should only be used when manual intervention is required, for Continuous Delivery deployments user input parameters should be avoided where ever possible.