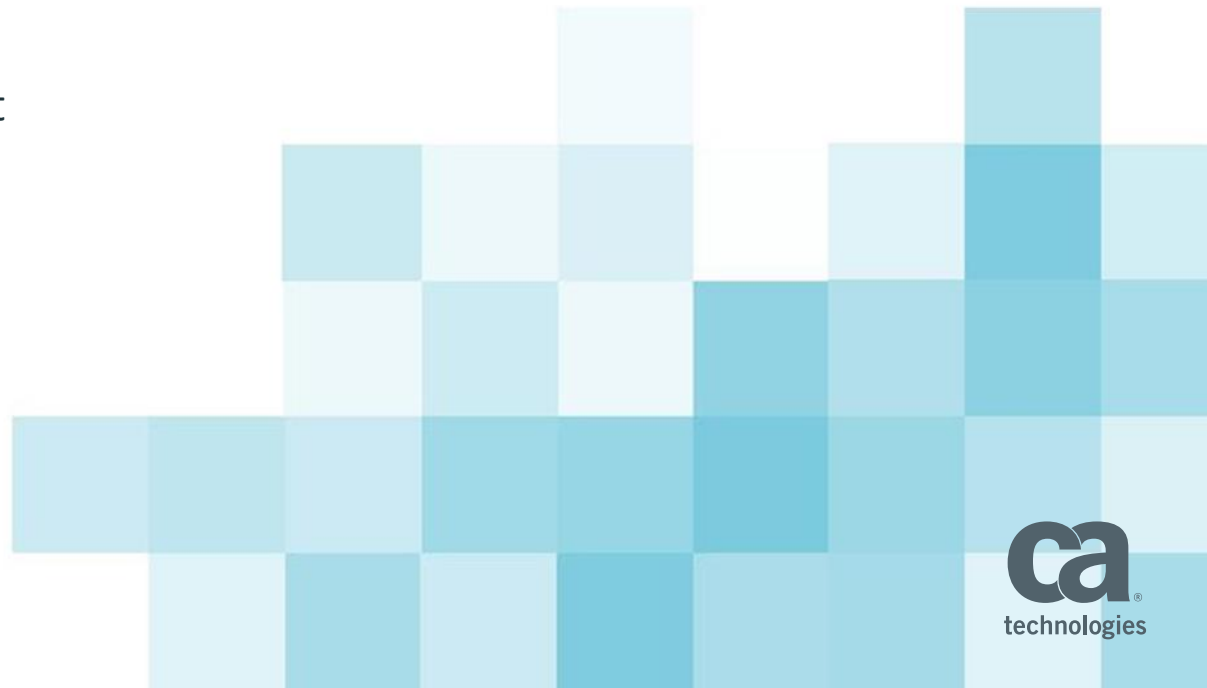


CA Release Automation Community Webcast Series

Release Automation
Shared Components

Walter Guerrero
Engineering Services Architect

October 5, 2016



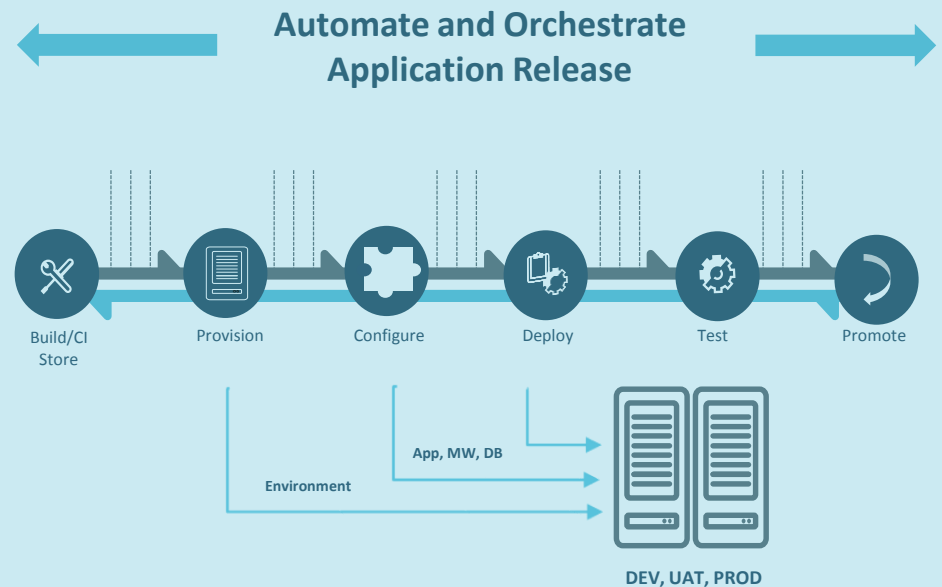
What We'll Cover in Today's Webcast

- What are Shared Components?
- Creating a Shared Component
- Shared Components Best Practices
- Shared Components Code Naming Conventions
- Troubleshooting Shared Components
- Export/Import Shared Components

Orchestrating the Continuous Delivery Tool Chain

PUSH-BUTTON SOFTWARE DEPLOYMENT ON-DEMAND

- Orchestrate an end-to-end continuous delivery tool chain
- Eliminate manual intervention by tightly integrating best-of-breed lifecycle tools
- Improve productivity and efficiency across all aspects of the lifecycle



CA Release Automation

Reliably and rapidly release applications on demand

CONTROL

Stabilize the release cycle and improve quality

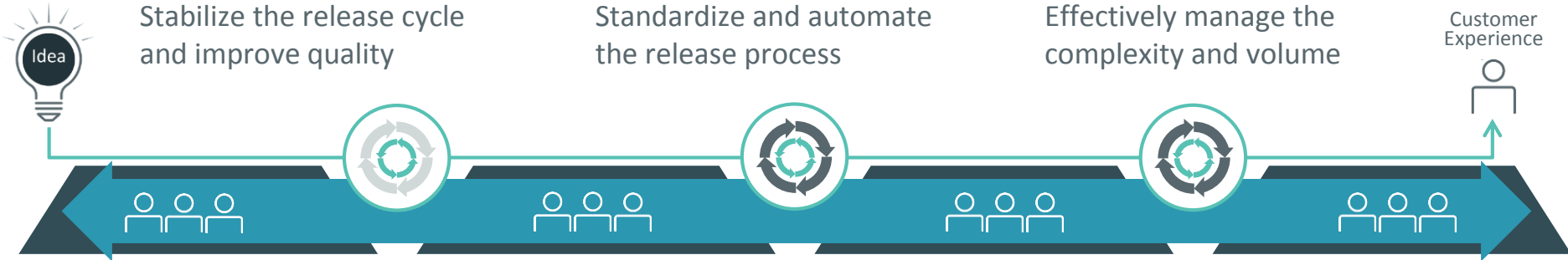
ACCELERATE

Standardize and automate the release process

SCALE

Effectively manage the complexity and volume

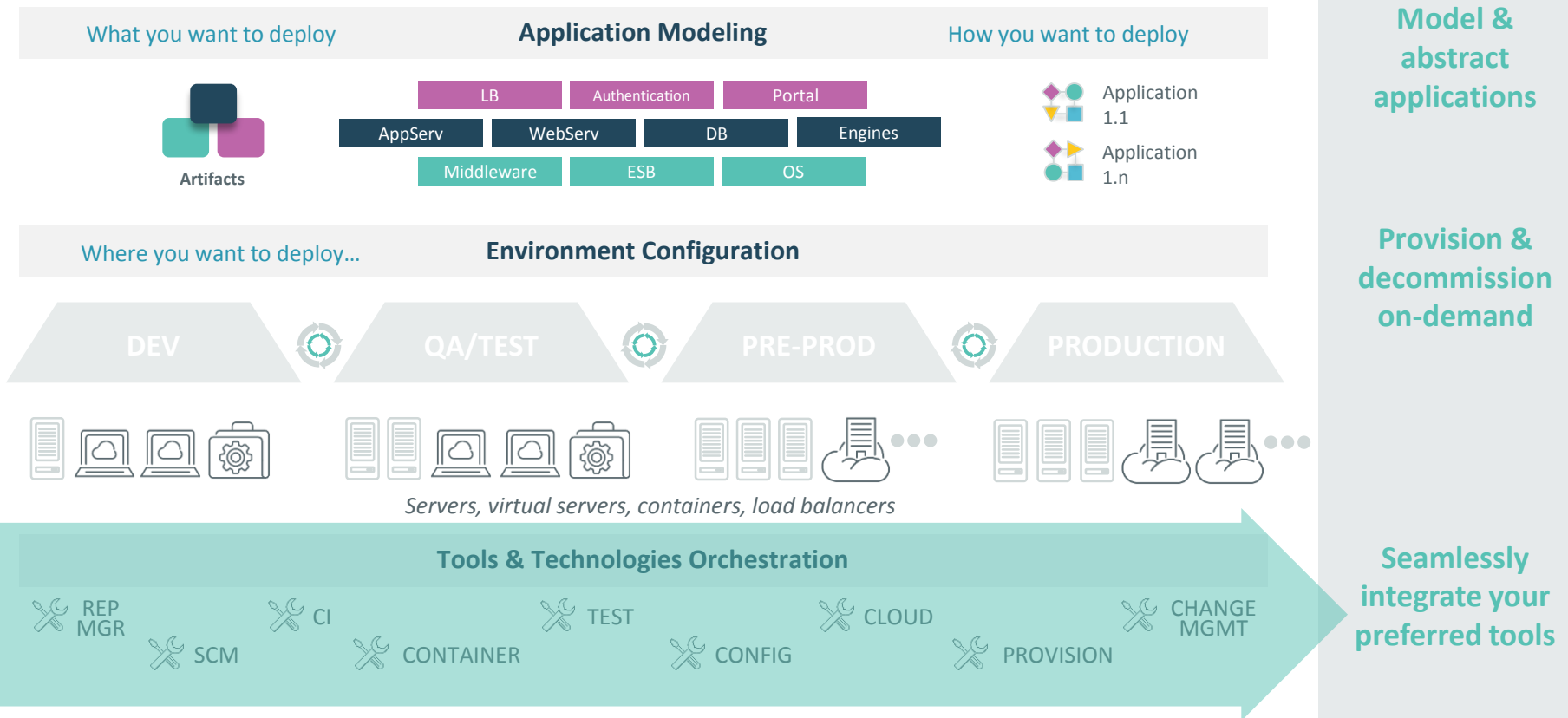
Customer Experience



“In my opinion, this is currently the best release automation tool in the market. It covers a lot of technologies and has made integration and adoption of different technologies and cross-platform deployments very easy.”

S&P 500 Electronics Company TVID: [A73-42C-9C7](#)

How Do You Harness Release Complexities?



What Are Shared Components?

Shared Components

What is a shared component?

- Shared components enable users to create deployment logic once and share this logic across multiple applications
- Each shared component has multiple revisions to maintain history of changes applied
- Shared components help organizations establish best practices by defining standards regarding common deployment logic

Shared Components

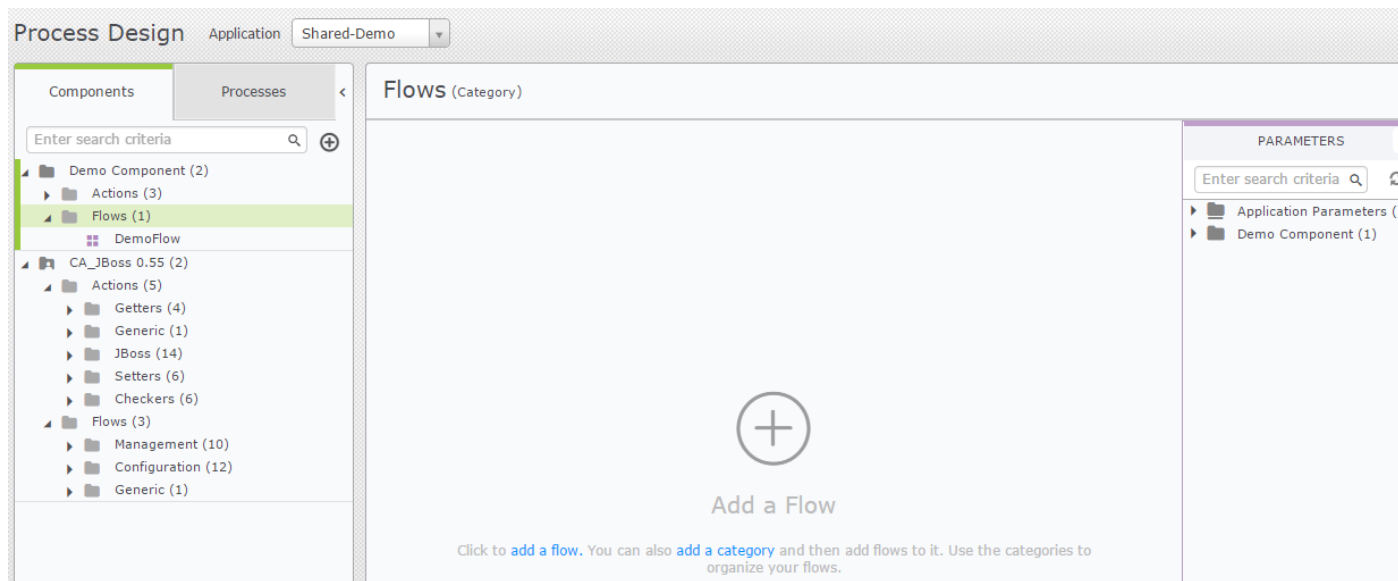
Using same guidelines

- Using shared components ensures that all teams follow the same guidelines and workflows.
 - Generic flows across multiple applications
 - Generic actions
 - Generic global artifacts
 - Coding naming standards

Shared Components

Typical view at Process Design

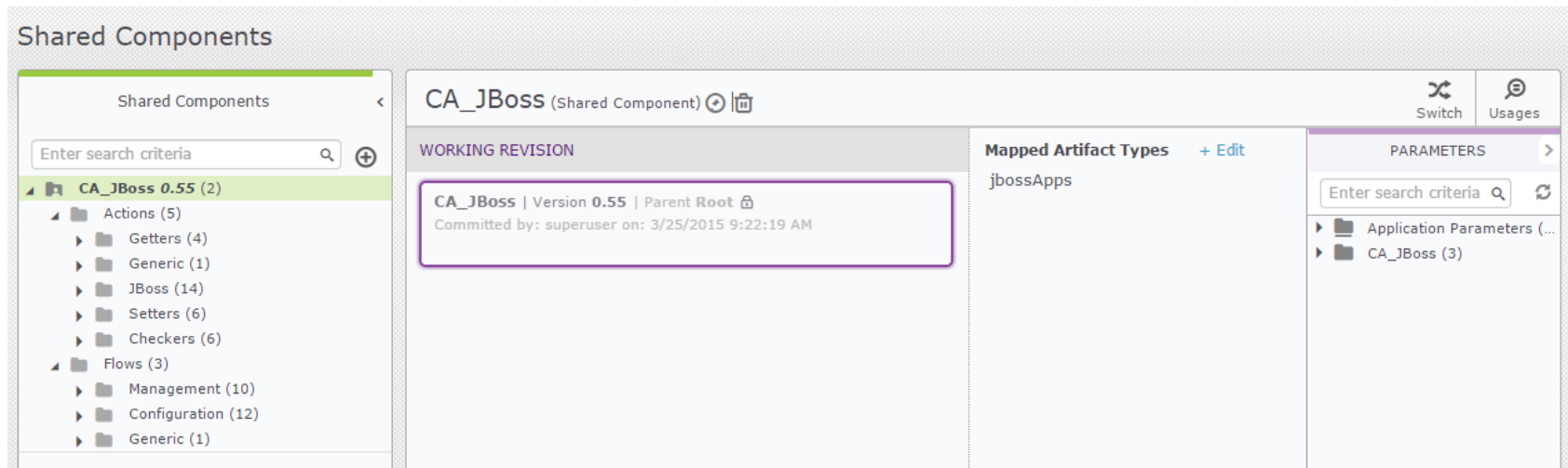
- A typical view of a shared components as part of an application component



Shared Components

Typical view at Shared Components view

- Typical view of the shared components in the Designer UI



Shared components can only be accessed via the ROC (Release Operations Console)

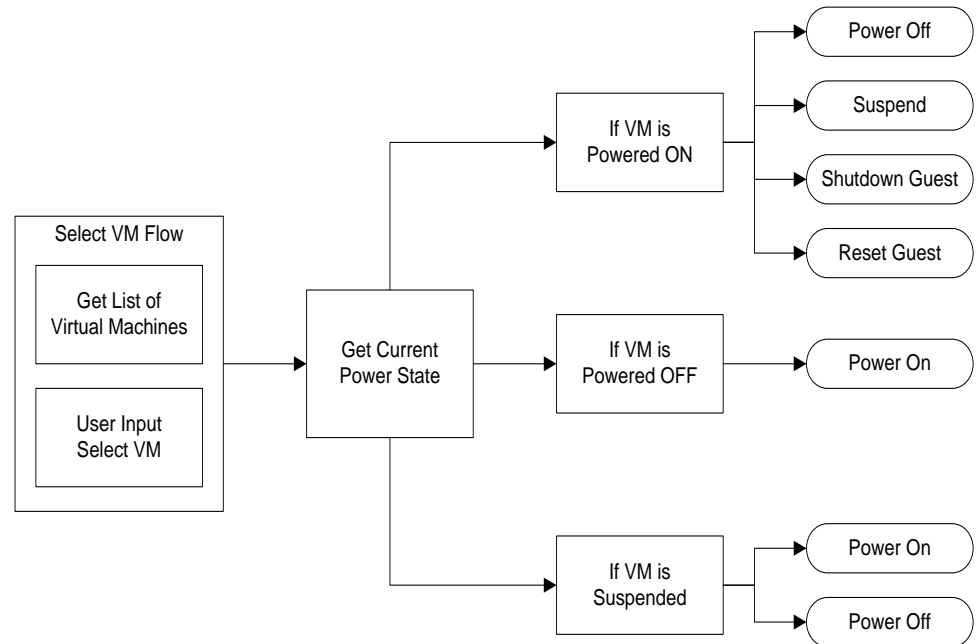
Creating A Shared Component

Creating A Shared Component

- The following steps are recommended
 - Step 1 – Shared component planning
 - Step 2 – Actions to be used
 - Step 3 – Flows to be used
 - Step 4 – Software Artifacts to be used
 - Step 5 – Server type assignment
 - Step 6 – Using shared components in applications

Creating A Shared Component Planning

- Create a flow chart with the different actions and their interactions
- Once the shared component has been created and a “release” version has been made available

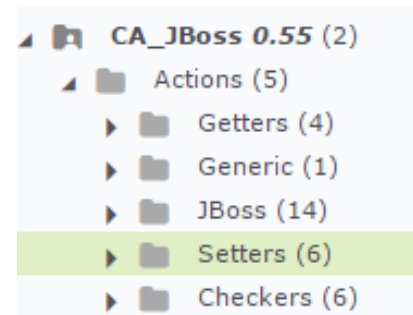


you will be able to build out processes based off these flows in an application within 10 minutes (based on complexity)

Creating A Shared Component

Actions to be used

- Look at the generic actions that you will need
- Select the correct action pack(s) that will be needed in the shared component.
- Under the “Actions”, define the following categories
 - Getters
 - Setters
 - Checkers
 - Generic
 - <SharedComponent>

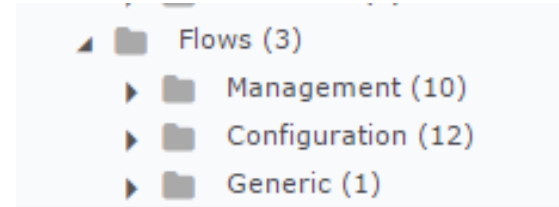


- Defined the parameters to be used
- Setup the environment and release variables to be used
- For shared components, need to add the parameters to the actions at this stage, since parameters cannot be updated in the flow for shared components

Creating A Shared Component Flows to be used

- Create categories for flows

- Generic
- Management
- Configuration

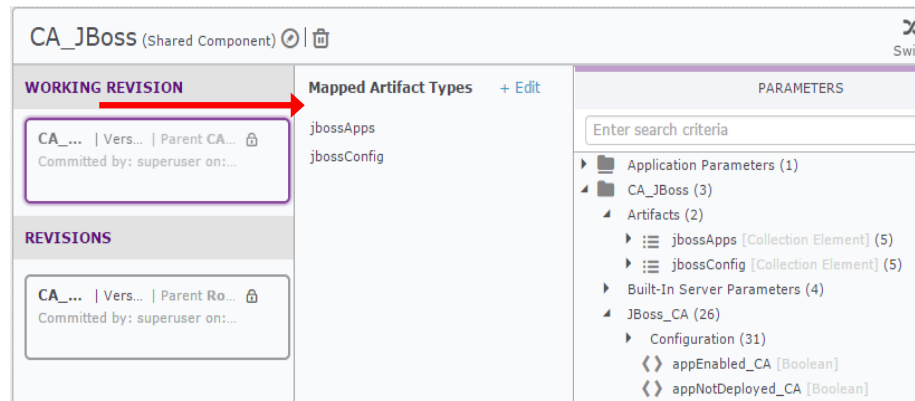


- Create additional categories as needed
- Create generic flows that perform commonly used tasks by the other flows
- For processing global artifacts create a sub-flow that will be using a loop based off the collection of a given artifact

Creating A Shared Component

Global artifacts

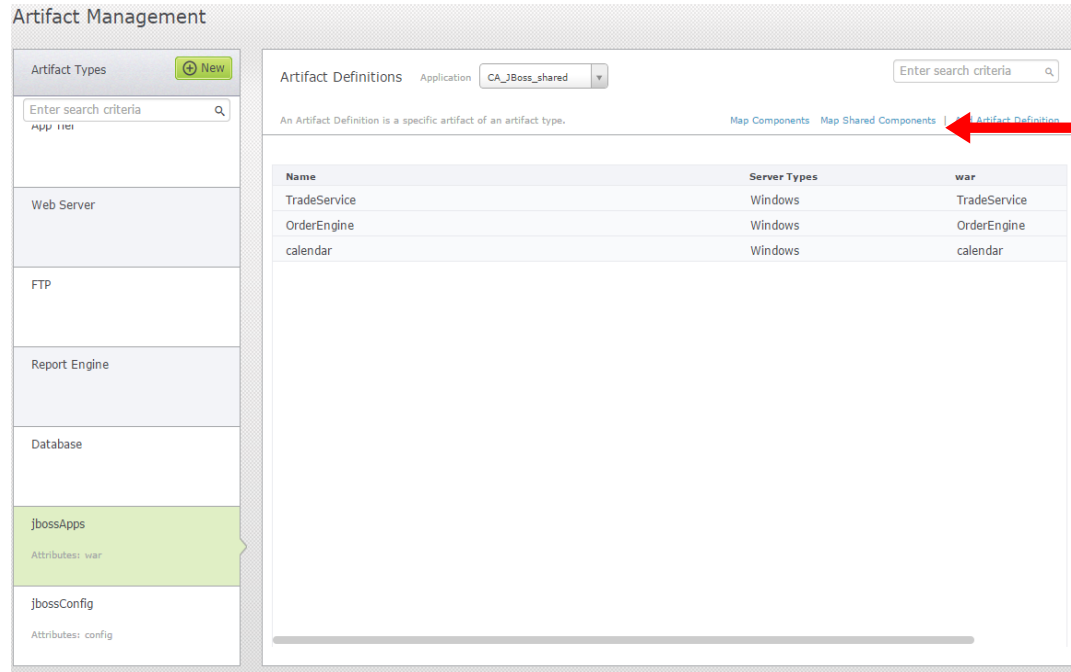
- A global artifact is an artifact that has been mapped to a shared component, where it can be ported to different applications in different Release Automation installations
- A global artifact can also be shared with multiple application at the standard component level
- The importance of global artifacts is its portability and the processing of the artifacts defined within the global artifact



Creating A Shared Component

Server types assignments

- Assign the server type to the global artifact definitions that will be used by the shared component



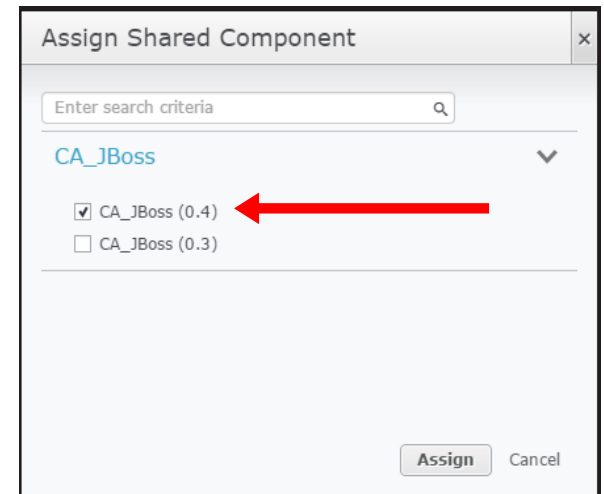
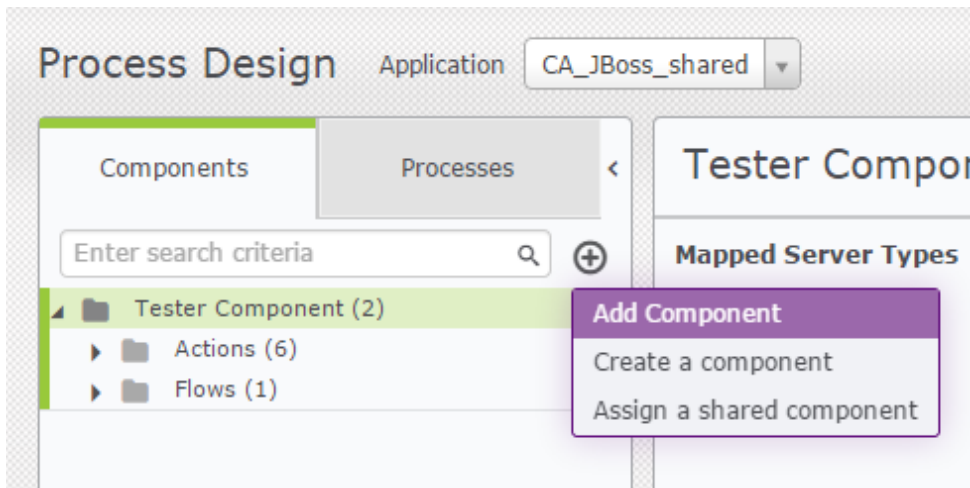
The screenshot displays the 'Artifact Management' interface. On the left, a sidebar lists 'Artifact Types' including Web Server, FTP, Report Engine, Database, jbossApps (highlighted), and jbossConfig. The main area is titled 'Artifact Definitions' for the application 'CA_Boss_shared'. It features a search bar and navigation links: 'Map Components', 'Map Shared Components', and 'Artifact Definition' (highlighted with a red arrow). Below these links is a table with three columns: 'Name', 'Server Types', and 'war'.

Name	Server Types	war
TradeService	Windows	TradeService
OrderEngine	Windows	OrderEngine
calendar	Windows	calendar

Creating A Shared Component

Using shared component in application

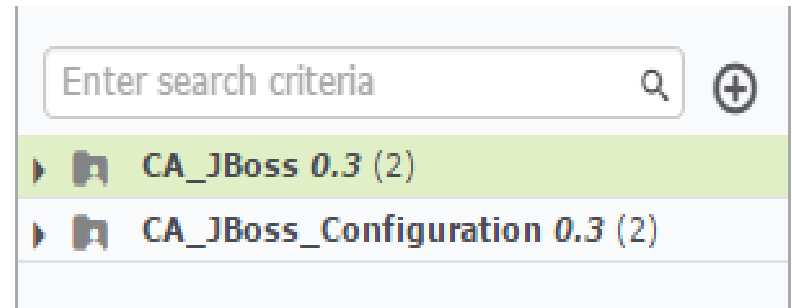
- Once the shared component has been created and imported (if necessary) into Release Automation, the working revision will become available to all applications in Release Automation
- In the design process, you will “Assign a shared component” to the application



Shared Components Best Practices

Shared Components Best Practices

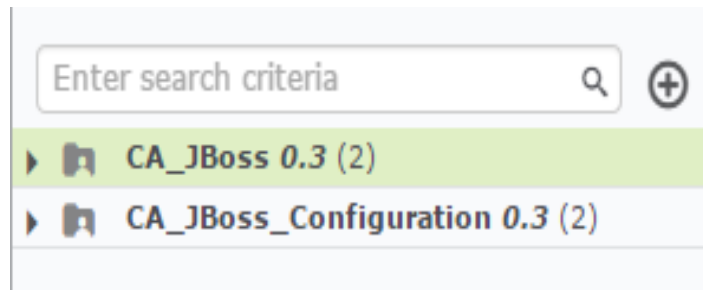
- These best practices will help you
 - Improve the quality of the shared component
 - Provide improved portability of the shared component



Shared Components Best Practices

Multiple revisions

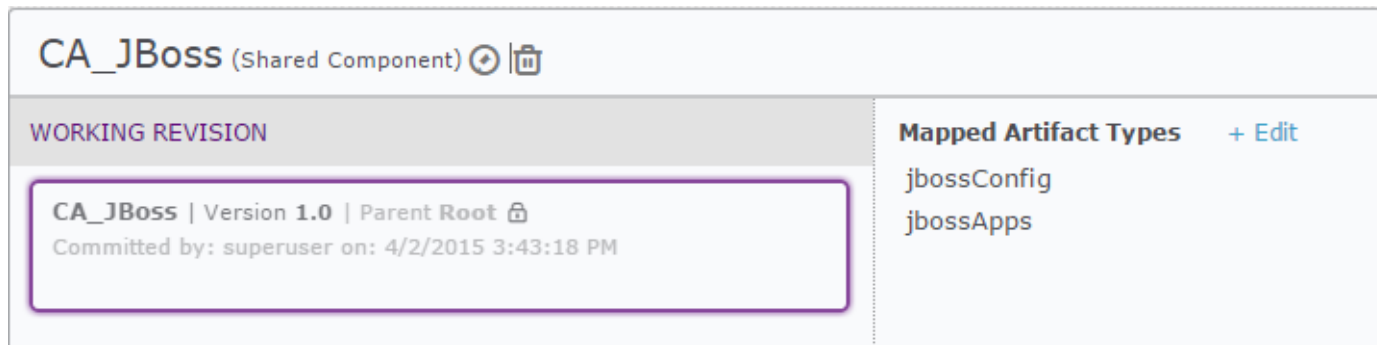
- To avoid confusion after you create the distributable zipfile of the shared component
- Remove all the older versions, and create a new distributable that contains just one revision (the latest revision)



Shared Components Best Practices

Mapped artifact types

- Use a global artifact type during the development phase of the shared component
- If the global artifact does not exist in the new RA installation, it will be automatically created
- Keep the name of the artifact type as generic as possible
- If you need a very specific artifact type, please list that in the release notes for the shared component



The screenshot displays the 'CA_JBoss (Shared Component)' interface. At the top, the title 'CA_JBoss (Shared Component)' is followed by a refresh icon and a trash icon. Below this, a grey bar indicates the current state is 'WORKING REVISION'. A purple-bordered box highlights the component details: 'CA_JBoss | Version 1.0 | Parent Root' with a lock icon, and 'Committed by: superuser on: 4/2/2015 3:43:18 PM'. To the right, the 'Mapped Artifact Types' section lists 'jbossConfig' and 'jbossApps', with a '+ Edit' link.

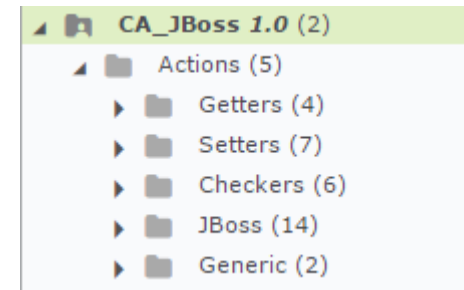
CA_JBoss (Shared Component)
WORKING REVISION
CA_JBoss Version 1.0 Parent Root
Committed by: superuser on: 4/2/2015 3:43:18 PM

Mapped Artifact Types + Edit
jbossConfig
jbossApps

Shared Components Best Practices

Component's actions

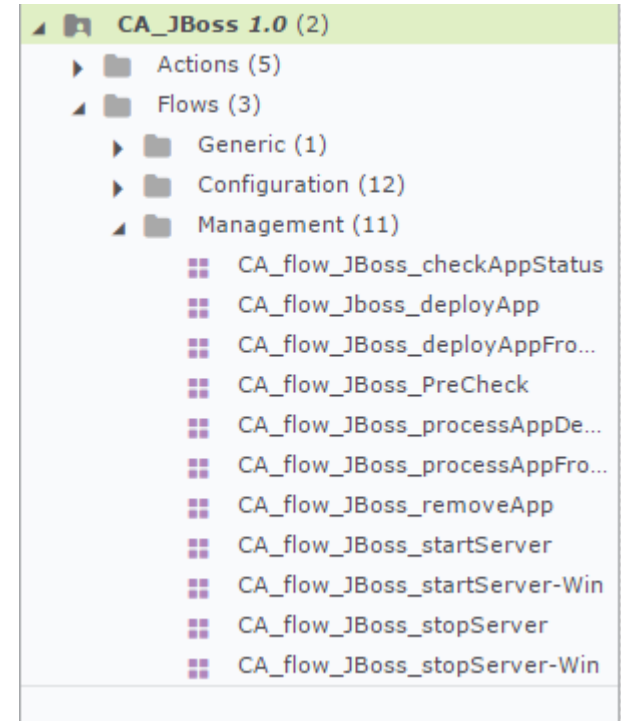
- To improve readability of the actions, create these categories
 - Checkers
 - Getters
 - Generic
 - Setters
 - <component's technology> → In this case, this will be i.e. JBoss



Shared Components Best Practices

Component's flows

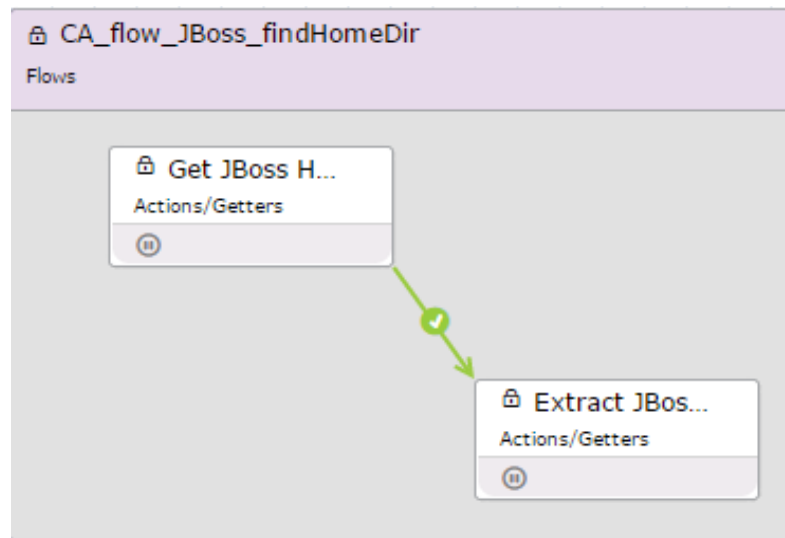
- The flows need to be named and setup based on the work to be performed
- Create the following categories
 - Generic
 - Configuration
 - Management



Shared Components Best Practices

Generic flows

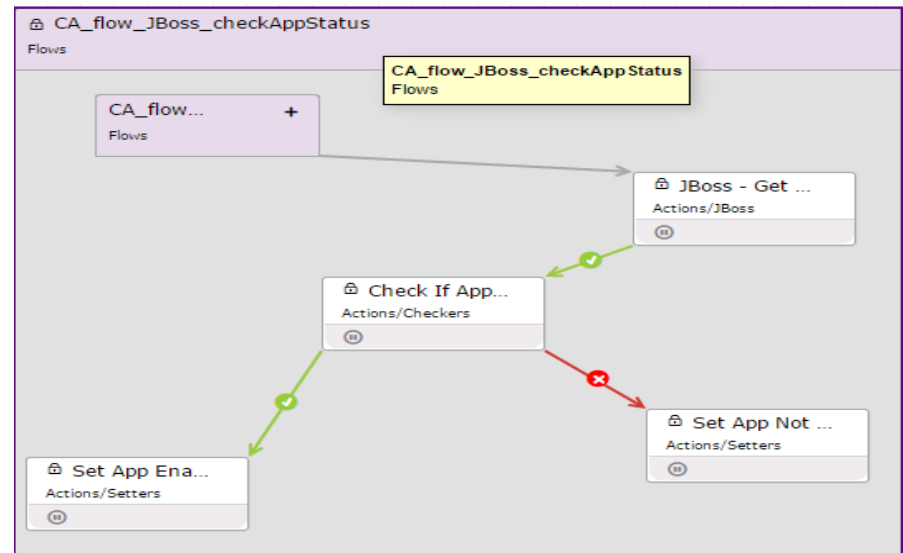
- Create generic flows that will be performing repetitive activities
- The information obtain by these generic flows needs to be placed in internal environment variables as needed



Shared Components Best Practices

Number of actions in flows

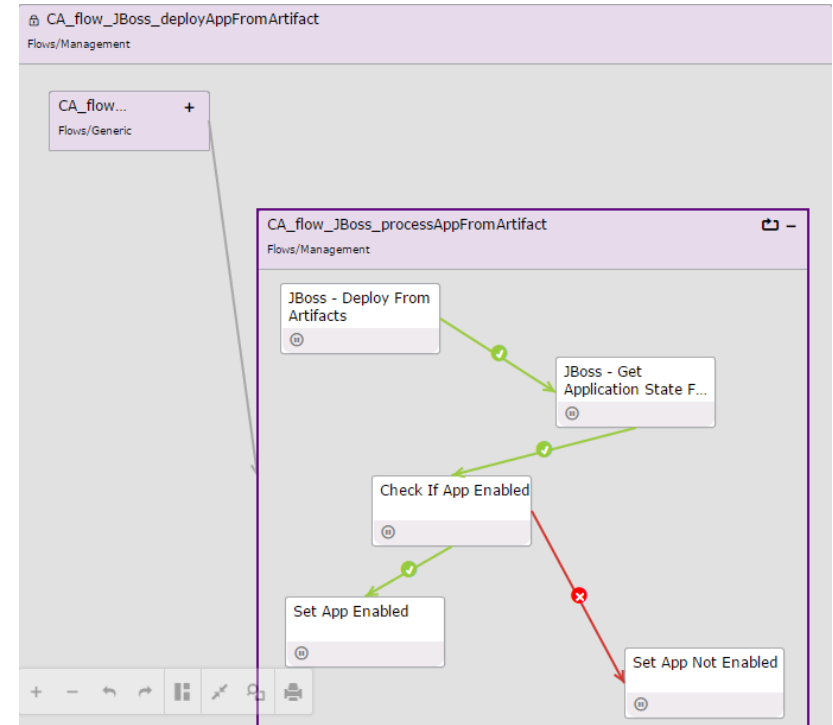
- The number of actions need to be limited to reduce complexity
- This can be accomplished by including other flows within a given flow



Shared Components Best Practices

Loops in flows

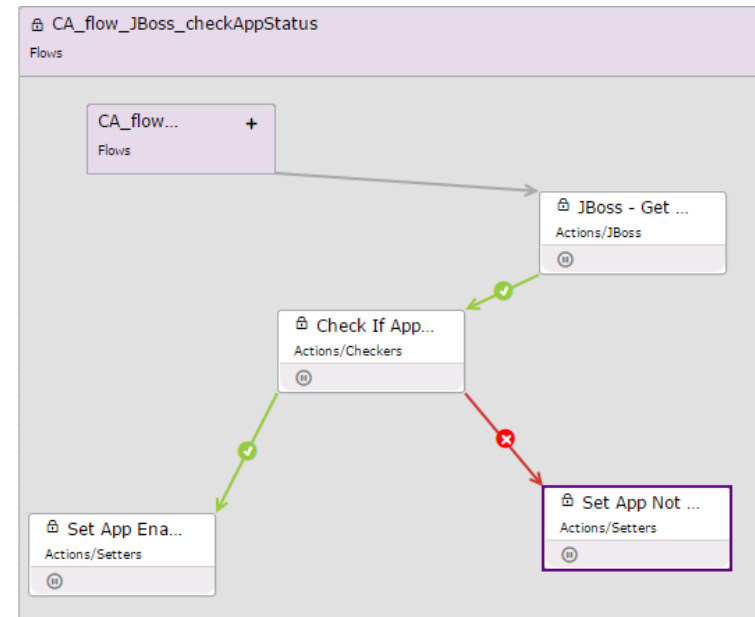
- Loops become very important when operating on global artifact types
- Flow needs to setup a “Collection” type loop for the global artifact types
- Create a generic flow that will loop thru the artifact type



Shared Components Best Practices

Settings Booleans as results of flows

- Setup Boolean values as a mechanism to define if the actions have completed successfully



Shared Components Code Naming Conventions

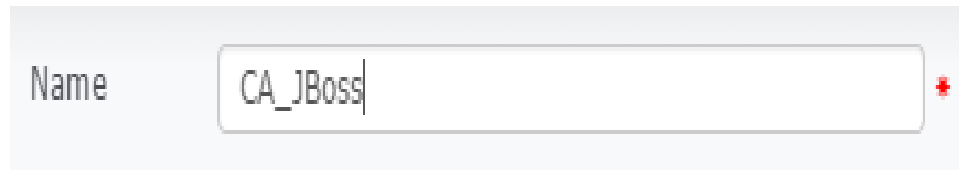
Code Naming Conventions

- It is important that **meaningful names** be used to name the components, flows, parameters, and collections that will make up the shared components libraries
- Use a **lower camel-casing**, where the first letter of each embedded word is lower case
- For the components and workflows, the names should start with “<CompanyName>_”. For parameters, this will become a suffix as shown “_**<CompanyName> <parameterScope>**”

Code Naming Conventions

Components names

- The component name should start with <Company_Name>
- For example, “**CA_**” for CA technologies provided shared components
- Add the technology for which this component is being written. i.e. “**CA_***Componentname*”



A screenshot of a web form with a label 'Name' and a text input field. The input field contains the text 'CA_JBoss' followed by a vertical cursor. A small red cross icon is visible to the right of the input field, indicating a validation error or required field.

Code Naming Conventions

Flow names

- The flow name should start with
<CompanyName_flow_Componentname_flowname>
- For example, “**CA_flow_JBoss**” for Jboss flows built by CA technologies

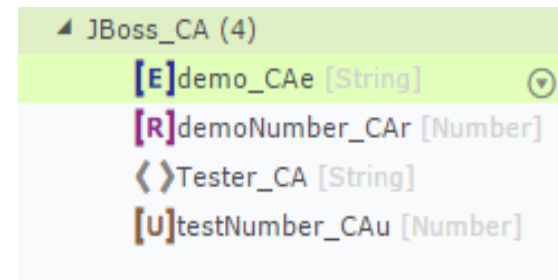


A screenshot of a web form with a label 'Name' and a text input field. The input field contains the text 'CA_flow_JBoss_start'. To the right of the input field is a small red error icon.

Code Naming Conventions

Parameters Names

- All parameters will end with the “_<CoName><parameterScope>” suffix; the scope will be based off the following scope values in Release Automation.
- Internal → no entry
- User Input → u
- Environment → e
- Release → r



Code Naming Conventions

Global Artifacts Names

- For the global artifacts that are part of the shared component is recommended that they be named as follows.
 - <sharedcomponent>artifact Type → jbossApps

Mapped Artifact Types

jbossApps

jbossConfig

Troubleshooting Shared Components

Troubleshooting Shared Components Revisions

- The revision number of a shared component is not automatically assigned
- Determine the best revision number sequence to use



A screenshot of a 'Branch Revision' dialog box. The title bar says 'Branch Revision' with a close button (X). The main text reads 'Create a branch from revision **CA_JBoss** version: 1.0'. Below this, there are two input fields: 'Version' and 'Description'. The 'Version' field contains the text '1.1' and has a red asterisk icon to its right, indicating a required field. The 'Description' field is empty. At the bottom right, there are two buttons: 'Branch' and 'Cancel'.

Troubleshooting Shared Components

Global Artifact Types

- The global artifact type needs to be mapped to the shared component
- The global artifact type also needs to be mapped to the application for the server type to be assigned properly

Artifact Definitions Application: Shared-Demo

An Artifact Definition is a specific artifact of an artifact type. [Map Components](#) [Map Shared Components](#) | [Add Artifact Definition](#)

Name	Server Types	Artifact Type
TestDemo	Server Type 1, Server Type 2	war

Mapped Artifact Types

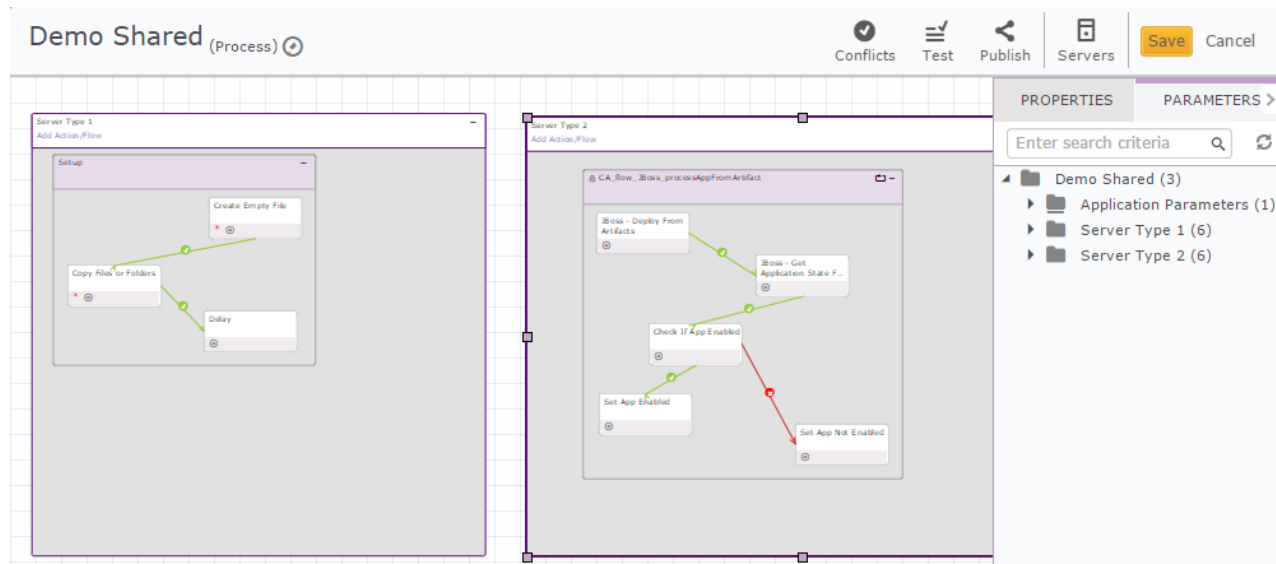
jbossApps

jbossConfig

Troubleshooting Shared Components

Adding Values/Parameters At Process Level

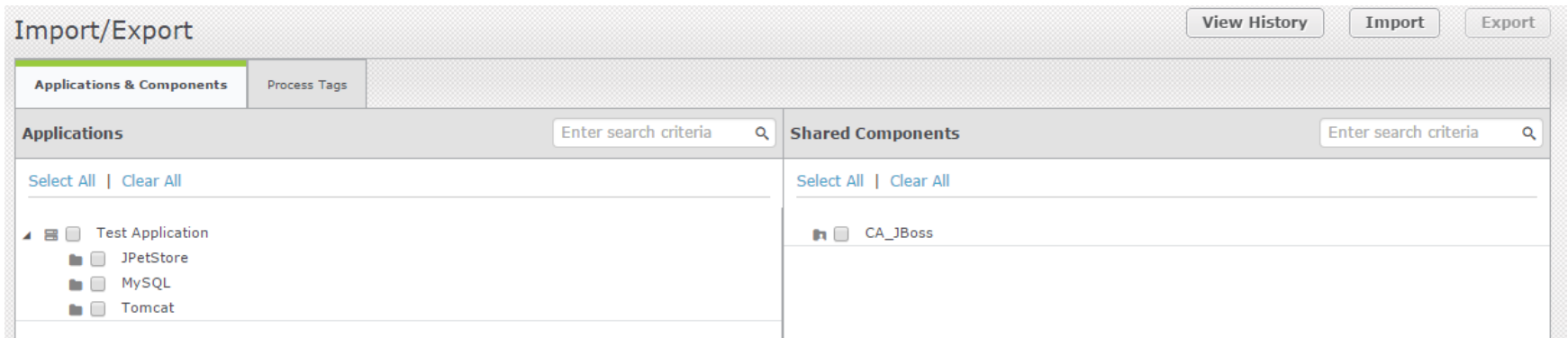
- Parameters and values cannot be added at the process/flow levels for shared components
- Values/Parameters can only be added/updated at the action level for shared component



Export/Import Shared Components

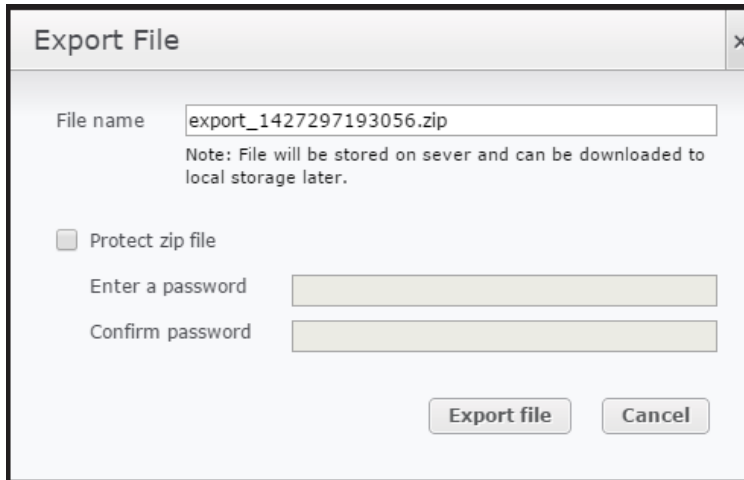
Export/Import Shared Component

- Shared components can be
 - Exported
 - Imported
- Provide the flexibility of separating development with production RA installations
- Password protect the shared component “zipfile”
- It may be necessary to include multiple shared components into a single zipfile package



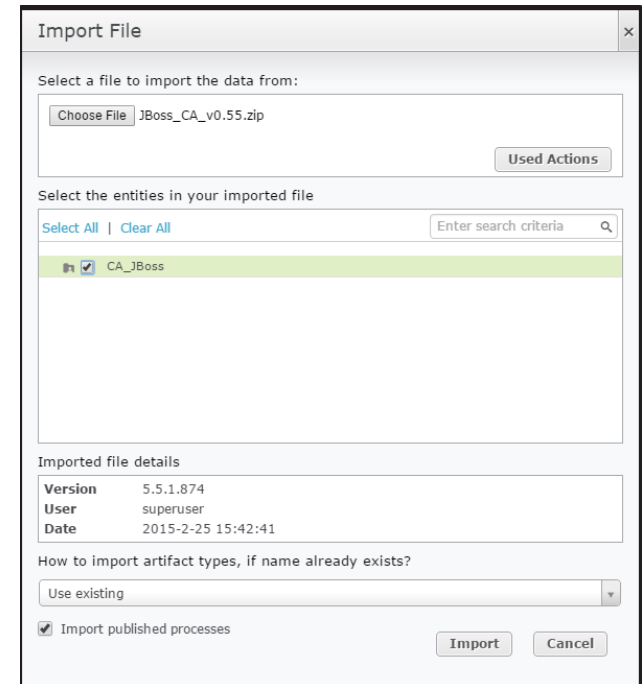
Export/Import Shared Component Wizard driven

- Export/Import is wizard driven
- Zipfile created resides in RA server, from which you can download



The 'Export File' dialog box contains the following elements:

- Title Bar:** 'Export File' with a close button (X).
- File name:** A text input field containing 'export_1427297193056.zip'.
- Note:** 'Note: File will be stored on sever and can be downloaded to local storage later.'
- Protect zip file:** A checkbox that is currently unchecked.
- Enter a password:** A text input field.
- Confirm password:** A text input field.
- Buttons:** 'Export file' and 'Cancel' buttons at the bottom right.



The 'Import File' dialog box contains the following elements:

- Title Bar:** 'Import File' with a close button (X).
- Select a file to import the data from:** A section with a 'Choose File' button and a text field showing 'JBoss_CA_v0.55.zip'. A 'Used Actions' button is to the right.
- Select the entities in your imported file:** A section with 'Select All' and 'Clear All' buttons, a search input field 'Enter search criteria', and a list box containing 'CA_JBoss' with a checked checkbox.
- Imported file details:** A table showing:

Version	5.5.1.874
User	superuser
Date	2015-2-25 15:42:41
- How to import artifact types, if name already exists?** A dropdown menu set to 'Use existing'.
- Import published processes:** A checked checkbox.
- Buttons:** 'Import' and 'Cancel' buttons at the bottom right.

Demonstration



Session Summary

- This session presented with the following information
 - You can create your own Release Automation shared components based on your business requirements
 - Shared components allowed you to setup coding standards for the generic components to be shared by your Release Automation's applications

Questions?

Comments?



In Closing

Questions after the webcast?

- Post them on the community site
- Contact me directly at walter.guerrero@ca.com



Walter Guerrero

Eng. Svcs Architect

walter.guerrero@ca.com



in