Testing a b4A Deployment Solution

Review of a b4A package management environment

best-blu consulting with energy GmbH

Your contact person:

E-mail:
Phone: +49 421 491 811 80
Date: May 27, 2025

Contents

Introduction

Deployment procedures based on *b4A Package Management* are essential for a functioning automation environment. It is therefore important that appropriate tests are defined and implemented when switching to such a procedure or updating the *best4Automic solution*. This must ensure that the processes and subtasks continue to function smoothly and cover the previously defined scope with the customer-specific configurations and extensions.

Important functions and topics that should definitely be covered in such tests are listed below. In addition, a few topics are mentioned that are only optionally necessary and are related to the respective configuration.

Finally, suggestions for implementation are described that can help with the automation of tests.

Test topics

The following topics for tests are divided into two categories. One part relates to the underlying *Automic Automation Engine* and the other part to the *best4Automic solution*. If tests from the first category are not successful, the other tests do not even need to be carried out.

Automic Automation

b4A Package Management is based on *Automic Automation* functions. For this reason, these functions should also be tested.

Technical user

The technical user used by *b4A* to access the *Automation Engine* is important.

- If the user has the necessary rights
    - Reading all object types?
    - Write/import all object types (in all target clients)?
    - Reading the tasks/activities (in all target clients)?

Import/Export

This function of the *Automation Engine* is the basis of the technical deployment process.

- Does the export of objects work on an XML basis?
    - The export must take place from the development environment
- Does the import of objects work on an XML basis?
    - If the versions of the stages are not exactly the same, the import should be tested in all stages

b4A Package Management

The following three areas of the test topics should be covered in all environments, as they relate to the basic functionalities.

b4A Package initialization

- Start the process for initializing a new *b4A package*
    - Has the *b4A package folder* been created in the correct base folder?
    - Have all defined subfolders been created?
    - Has the metadata variable been created and does it contain the correct entries?

Build *b4A Package* Release

- Start the process for building a release
    - Has a release archive been created?
    - Does the release archive contain all the necessary files?
        - Are all object exports included?
        - Are configuration objects (including environment and client-specific variants) included?
        - Does the file ".package.info" exist?
- Compliance Check
    - Have all standard tests been carried out and have errors been detected?
    - Have all customer compliance tests (CCTs) been carried out and have errors been correctly identified?
- Dependency Verification
    - Were the dependencies to other *b4A packages* recognized correctly?

Installation *b4A Package* Release

- Start the process to install a *b4A package* release
    - Have all objects been imported correctly?
    - Have all configuration objects been imported?
        - Have the correct variants been imported?
        - Were runtime objects imported if they did not previously exist or were they not changed if they already existed?
    - Have obsolete objects been moved correctly?

Package Index

If the *b4A Package Index* is used, the correct functioning should be checked.

- Initialization of a *b4A package*
    - Has the *b4A package* been included in the index?
- Building a *b4A package* release
    - Has the index entry been updated accordingly?
- Installation of a *b4A package* release
    - Has the index entry been added (when installing the *b4A package* for the first time) or updated (when updating the *b4A package*)?

Git integration

If the *b4A Git integration* is used, the correct functioning should be tested.

- Technical user or access token?
    - Are the correct accesses enabled for the project?
        - Creating Git repositories in the project
        - Push to all Git repositories of the project
        - Pull from all Git repositories of the project
- Initialization of a *b4A package*
    - Has the Git repository been created?
- Building a *b4A package* release
    - Have the sources in the Git repository been updated in the correct Git branch?
- General
    - Can Git branches be created?
    - Is the difference between the *Automic Automation Engine* in the development client and a Git branch correctly analyzed?

Implementation

How exactly these tests are implemented is up to each customer. For some, manual testing is the best option and for others, partial automation. Some may go in the direction of full automation. Below are a few suggestions for possible automation of such tests.

Prerequisites

At least two clients are required to test a deployment procedure. The first client represents the development environment and the second client a possible target for the installation. A third client, as a further installation target, can extend the possibilities for tests.

Partial automation

All tests can be carried out and checked manually, but this takes a lot of time. It is therefore advisable to automate these processes as far as possible. Two tools are available for this purpose, which can be combined well.

Automic Automation

All test executions can be implemented using workflows in the *Automic Automation Engine*. This also applies to checking the correct results of the tests and preparing the test environment.

b4A Test Automation

At the same time or as a supplement, *b4A Test Automation* can also be used to further increase the degree of automation and to formulate the tests in a standard language for test definitions at (Cucumber) . In addition, the test results are automatically documented and can thus be further processed directly.