



blueprism

The Operational Agility Software Company

Blue Prism

Java Automation Guide

The information contained in this document is the proprietary and confidential information of Blue Prism Limited and should not be disclosed to a third party without the written consent of an authorised Blue Prism representative. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying without the written permission of Blue Prism Limited

© **Blue Prism Limited**

All trademarks are hereby acknowledged and are used to the benefit of their respective owners.

Published by:

Blue Prism Limited
Centrix House
Crow Lane East
Newton-le-Willows
WA12 9UY, UK
Registered in England; Reg. No. 4260035
www.blueprism.com
Tel: 0870 879 3000

Contents

1	Introduction	4
2	Installation & General Mitigation	4
2.1	Java Access Bridge (JAB)	4
2.2	Launching and Attaching	5
2.3	Best Practice	5
3	Interfacing with Java Applications	7
3.1	Wait Stages	7
3.2	Navigator	7
3.3	Application Wizard Options	8
3.4	Java Element Attributes	8
4	Troubleshooting	10
4.1	Target Application Crashes	10
4.2	Navigation stages not working or causing Blue Prism to hang.....	10
4.3	The spy tool doesn't recognise elements	10
4.4	Attempting to write to a text box produces an error message	10

1 Introduction

The purpose of this document is to act as a useful source of advice and various helpful tips to Blue Prism developers who are attempting to automate Java applications. This document can be referred to when attempting to troubleshoot issues encountered when automating Java applications.

2 Installation & General Mitigation

In this chapter we will discuss some useful tips regarding installations and general setup for automating Java applications, as well as advice for launching and attaching to Java applications.

2.1 Java Access Bridge (JAB)

The Java Access Bridge is a technology that exposes accessibility information about Java applications and applets to external applications such as Blue Prism.

Since Java 1.7 Update 6, the JAB is built into Java, meaning JAB simply needs to be enabled by ticking the "Enable Java Access Bridge" check-box found in the Control Panel - Control Panel\Ease of Access\Ease of Access Center\Make the computer easier to see. Steps to reach this (using Windows 7):

1. Go to Start > Control Panel > Ease of Access > Ease of Access Center. Alternatively, press Windows logo key+u to access the Ease of Access Center.
2. Select Use the computer without a display.
3. In the section Other programs installed, select the check box Enable Java Access Bridge (you may have to scroll down)

Other programs installed

These programs are available on this computer. Running more than one at a time might cause conflicts.

Java Access Bridge, from Oracle, Inc. providing Assistive Technology access to Java applications

Enable Java Access Bridge

Enabling Java Access Bridge

Un-ticking the check-box does not actually disable the JAB as you would imagine. To disable JAB, you will need to run the following command in command prompt, where %JRE_HOME% is the location of the JRE on the machine:

```
%JRE_HOME%\bin\jabswitch -disable
```

The JAB installer found on the Blue Prism Portal is intended for use on machines that have an older version of Java that does not include the JAB as standard. Java automation requires the JAB to be installed, which is available here:

<http://www.oracle.com/technetwork/articles/javase/index-jsp-136191.html>

At the time of writing, the latest available version of the JAB is 2.0.2. Blue Prism supports version 2 and 2.0.2. Version 2.0.1 contains a critical fault which causes problems with Blue Prism, so it should be avoided. The Java Runtime Environment (JRE) is also required. It is recommended to use version 1.2.2 or later with Blue Prism. If you have an earlier version installed, contact Blue Prism support.

There are two methods of installing JAB: automatic and manual. The former is preferred and will usually suffice in the majority of situations but a manual installation may be required if problems using automatic installation are encountered, for example when installing for non-standard JREs.

For a detailed guide on how to install JAB for use with Blue Prism, check the Java Access Bridge Installation Guide located within the Blue Prism help files.

It is crucial to remember the following:

- Each time the JRE is upgraded or modified (for example for a security update issued by Oracle, the developer of JRE), the JAB installation will have to be repeated in order to ensure that the new JRE has the correct extensions in place to work with the Access Bridge.
- Each time a new JRE is installed on the target machine (for example for use with a new piece of software), it may be necessary to repeat the installation of the JAB, even for existing JREs.

2.2 Launching and Attaching

It may be that the traditional method of launching the target Java application you wish to interface with involves running an external script or something similar. This might be used so that the script ensures that the application uses a specified version of JRE (possibly an older version which is installed side by side with other versions), or for other custom reasons. Because of this you might need to launch the Java application via the command line and then attach to it from within your Business Object.

2.3 Best Practice

The following tips should be followed when running Java Applications which are being automated by Blue Prism:

- Ensure that you are using Blue Prism version 4.1 or later.

- Restart the target Java application at regular intervals.
- Restart the resource PC every 24 hours

Adhering to the above should reduce the number of issues commonly encountered.

3 Interfacing with Java Applications

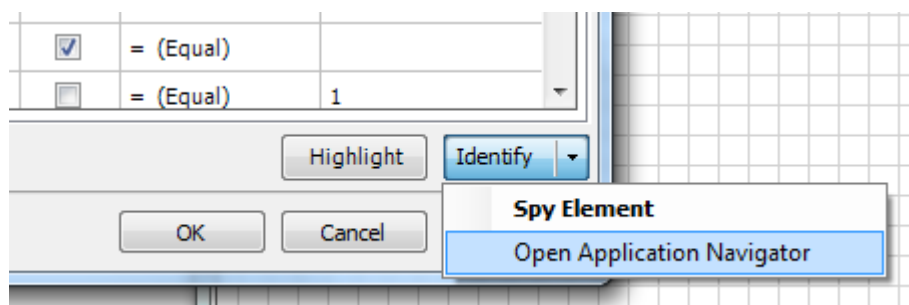
3.1 Wait Stages

Using a Wait stage to check if a Java element exists requires the Java Application Model tree to be traversed. Constantly accessing the model can be costly in terms of CPU efficiency, so it is advisable to adhere to the following tips:

- Avoid using Wait stages that continually check for the existence of a Java element. A ten-second wait stage would thrash the application model considerably. An alternative would be to use a Wait stage with a fixed time period before the Wait stage that checks whether the element exists. By doing this, sufficient time should have passed to allow the element to appear, meaning that the first check of the application model should suffice.
- Avoid using Wait stages that check for several conditions at once. For example, there is not much point checking for an element that rarely occurs, especially when doing so could result in potentially costly CPU instructions. It would be better to wait for a lesser number of conditions, tiered by likelihood of occurring. You could always loop back to this Wait stage if it times out.
- When waiting for Popups or other dialog windows, it is best to spy these windows using Win32 mode. That way, there is no need to access the Java Application Model tree when checking for their existence.

3.2 Navigator

If you encounter any unexpected problems that prevent you from using the any spying mode, an alternative exists in the form of the Application Navigator. To open the navigator, you need to click on the drop down arrow next to the Identify button in Application Modeller and select Open Application Navigator.



Opening Application Navigator

Once opened, the application navigator will scan the Java Application Model Tree for all the elements it can find. It will then display a tree structured list of Java elements from which you

can choose from to create your elements in Application Modeller. You will notice that when you select an element in the tree it will be highlighted within the application for confirmation and its attributes will be displayed.

3.3 Application Wizard Options

When using the Application Wizard while creating your Business Object you will notice that there is an option to add some options, possibly as instructed by Blue Prism support. These options may be set to modify the behaviour of the interface with the target application or to enable special features. They can be added in a comma-separated list as required. Two options exist for Java applications:

- **descendtree:** This option causes Blue Prism to search the entire Application Model Tree for elements within the target Java application. If this option is not set, Blue Prism will only directly retrieve a list of visible Java elements for each visible window in the target application. Descending the tree is usually slower, but some target applications and Java Runtime Environments can crash when using the default method.
- **ignorenotshowing:** This option will tell Blue Prism to ignore elements in the Application Model Tree which are below elements which are reported to be not showing. Using this option may result in a significant performance improvement.

3.4 Java Element Attributes

As is the case when dealing with other kinds of target applications, it can be considerably advantageous to use some Java element attributes. Enabling the “descendtree” option may be required to enable some of the following attributes:

- **MatchIndex:** The match index attribute allows the search to stop as soon as the element Nth element is found. This should be used carefully because usually the search would continue looking for duplicate matches and thus eliminates an ambiguous match. Using this attribute to match elements can result in significant performance gains by reducing the scope of a query to the target application.
- **MatchReverse:** The match reverse attribute simply changes the direction of the search, this can be useful if the element is at the end of the search list. The match reverse attribute is only relevant when used in combination with the match index attribute.
- **AncestorCount:** The ancestor count attribute is particularly useful in a tree like object model which includes Active Accessibility and Java. Within these models searches along the branches of the tree can be stopped short when we are about to descend deeper than the ancestor count. Ancestor count can be used for all application types but only provides optimisation in AA and Java.

- **Showing:** This attribute indicates the element, the element's parent, the element's parent's parent, and so on, are all visible. Note that this does not necessarily mean the element is painted on the screen; it might be occluded by some other showing element. Using this attribute can speed up interfacing with certain Java elements.

When using “descendtree” mode as described in chapter 3.3, significant performance gains can be achieved by using the AncestorCount, MatchIndex and MatchReverse identifiers to limit the scope of a query.

4 Troubleshooting

4.1 Target Application Crashes

If elements spy correctly, but then the target application crashes when attempting to highlight or otherwise interact with those elements, the problem is likely to be caused by a flaw in the Java Access Bridge components, which have problems with large numbers (approximately 200 or more) or elements on a single screen. You can prevent this crashing by enabling the “descendtree” target application option in Application Modeller, as described in chapter 3.3. This mode is significantly slower, so should only be used when necessary.

4.2 Navigation stages not working or causing Blue Prism to hang

Previously there have been issues when using a navigate stage to select a Java element such as a button or checkbox that cause Blue Prism to hang. This problem has since been fixed in later versions of Blue Prism so ensure that you are using at least version 4.1.

If a Navigate stage fails to set the value of a checkbox or radio button, you can use a write stage as an alternative, you will need to send the value “True” (without quotes) to the element. This should result in it being selected.

4.3 The spy tool doesn't recognise elements

If the spy tool does not seem to recognise elements as you hover over them, check the following common causes first:

- Failing to reboot the target machine after installing Java Access Bridge.
- Using the wrong Java Access Bridge version.

4.4 Attempting to write to a text box produces an error message

An error message such as “Call to setTextContents failed. This method is known not to work against version 1.3.x of the Java Runtime Environment” indicates that you are trying to write to a Java text box in an application running under JRE 1.3. Unfortunately, this is impossible due to a missing implementation in the Java Access Bridge connectors that correspond to that JRE version.

In all cases there is a workaround, though some thought and ingenuity may be required on the part of the user. Here is a list of suggested workarounds for common problems:

- **General Workaround - upgrade the Java Runtime Environment (JRE).** In most cases your target application should continue to function exactly as before following an upgrade, but you should consider the associated risk of any potential problems before doing so. Blue Prism will function reliably against version 1.4.0 or greater.
- **To work around a writing problem -** If you are having a problem writing data to an element, you may send keystrokes to the application instead. This involves activating the window containing the element of interest, focusing the element ready to receive keyboard input (for example by clicking on it or tabbing to it), and using a Navigate stage to send keystrokes. Such keystrokes may need to include backspace or delete in order to clear any existing data in the element.