**Adaptive Strong Authenticator**
Configuration Guide
10*g* (10.1.4.3.0)

December 2007

Adaptive Strong Authenticator Configuration Guide, 10g (10.1.4.3.0)

# Contents

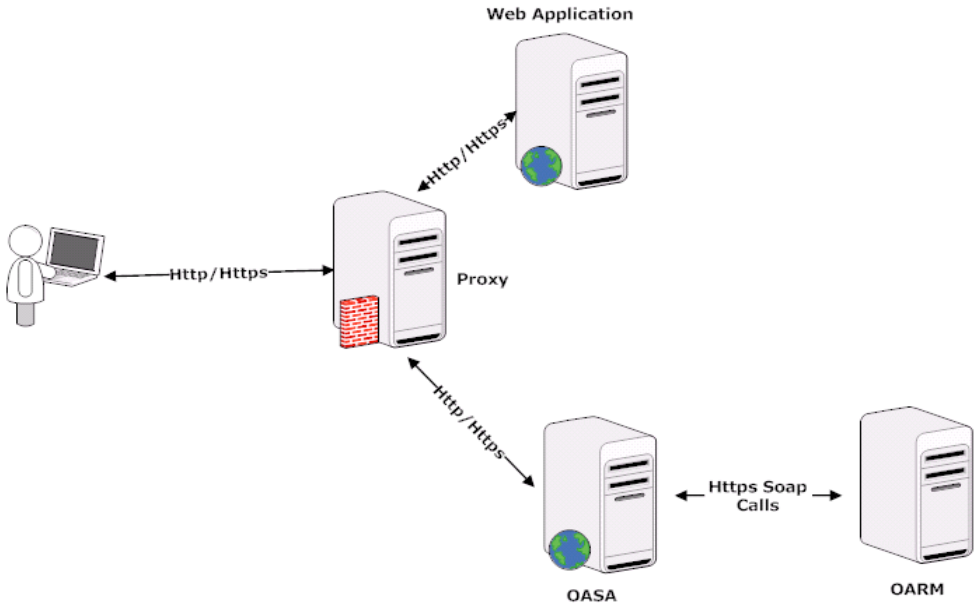
# Introduction

The Adaptive Strong Authenticator Configuration Guide provides information on customizing the client-facing Adaptive Strong Authenticator web application. The Oracle Adaptive Access Manager's Universal Installation Option (UIO) offers multi-factor authentication to web applications without requiring any change to the application code. The Adaptive Strong Authenticator configuration is specific to the UIO deployment. Please refer to the architectural diagram below for the components involved.

The user interface provided by the Adaptive Strong Authenticator web application can be easily customized to achieve the look-n-feel of the customer applications. This document is intended for integrators who install and configure Adaptive Strong Authenticator to support one or more web application authentication and user registration flows.

# Architecture

The following diagram shows an Adaptive Risk Manager UIO deployment.




The Adaptive Strong Authenticator proxy intercepts the HTTP traffic between the client (browser) and the server (web application) and performs appropriate actions, such as redirecting to the Adaptive Strong Authenticator, to provide multi-factor authentication and authorization. The Adaptive Strong Authenticator in turn communicates with Adaptive Risk Manager to assess the risk and takes the appropriate actions, such as permitting the login, challenging the user, blocking the user, and other actions.

# Settings

The Adaptive Strong Authenticator configuration is controlled through properties files.

## Configuration files

The following files are used to configure the Adaptive Strong Authenticator

- The `bharosauio_client.properties` file contains the client-configured properties (any properties that have been customized for a specific deployment). These client-configured properties will override the default configurations contained in the `bharosauio.properties` and `bharosauio_msg.properties` files.
- The `bharosauio.properties` file contains the default UIO system /device configurations. The file deals with the structural changes in the overall application. It is where the header, footer, and CSS properties are located.
- The `bharosauio_msg.properties` file contains the default UIO messaging and page content configuration. For example, page titles, links at the bottom of the pages, page messages, error message, and confirmation messages.

# First Steps

The first steps of Adaptive Strong Authenticator configuration and customization are:

1) Determine the application ID of each application being secured.
2) Assign default user groups for each application being secured.

## Application ID

UIO can be placed in front of multiple applications, and customized to work with each one as required. Determine how many applications are going to be configured, assign each application an Application ID. This Application ID should be the same one used to configure the Proxy (see the *Oracle Adaptive Access Manager Proxy Integration Guide*).

The Proxy will send the `AppId` to the Adaptive Strong Authenticator as needed via HTTP header. This `AppId` is then used to determine which configuration is used when displaying pages to the client. Adaptive Strong Authenticator is configured by a set of properties which we will discuss in more detail later. For now, here is an example of how `AppId` is used in a property definition.

```
bharosauio.appId1.default.user.group=app1Group
```

The bold "**appId1**" is the location in the property where the `AppId` is used to configure application specific values.

## Default User Groups

Each application can be configured to have a unique default user group. This is the group that a user of that application will be associated with as their primary user group when first created in the Adaptive Risk Manager Online database. Similarly, it will be the group used to attempt to load user information from the database when a user attempts to log in to the application.

As used in the previous example the property for default user group looks as follows:

```
bharosauio.appId1.default.user.group=app1Group
bharosauio.appId2.default.user.group=app2Group
```

In this case you can see that we have defined two user groups to two different applications. The application with an `AppId` of "`appId 1`" has been assigned the default user group of "`app1Group`" and the application with an `AppId` of "`appId2`" has been assigned the default user group of "`app2Group`".

# Branding

The Adaptive Strong Authenticator user interface branding is customized in several ways.

1) Custom header / footer files
2) Custom CSS file
3) Custom properties for page content and messaging

## Custom Header / Footer

Adaptive Strong Authenticator provides the ability to create a custom header and / or footer file for applications being secured. The header and footer files are JSP and can contain any HTML or JSP code required to replicate the look of the application being secured. All the customer resources (JSP files, image files, HTML, and others) should be deployed along with the Adaptive Strong Authenticator web application.

The header and footer files should contain only content html, all page related tags (<html>, <head>, <body>, etc) are already provided by the Adaptive Strong Authenticator. As a simple example we will create a header and footer that contain a single image each, to be used as the header and footer of an application called "`appId1`".

*/client/app1/header.jsp*

```
<img src="/client/app1/images/header.jpg" alt="Welcome to App1"/>
```

*/client/app1/footer.jsp*

```
<img src="/client/app1/images/footer.jps" alt="App1 Footer"/>
```

To associate these files with the application we would add the following properties to `bharosauio_client.properties`:

```
bharosa.uio.appId1.header = /client/app1/header.jsp
bharosa.uio.appId1.footer = /client/app1/footer.jsp
```

## Custom CSS

The Adaptive Strong Authenticator styles are controlled through a single CSS file (bharosa_uio.css). These styles can be overridden by including a custom CSS file. Much like the header and footer example above, you can create your own file and include that file on an application or global level through properties (see "How the Properties Work" in this document).

In this example we will override the font-family of the default body style definition.

The body style in bharosa_uio.css is defined as follows:

```
body{
        background-color:#ffffff;
        font-size:12px;
        color:#000000;
        font-family:arial,helvetica,sans-serif;
        margin:0px 0px 0px 0px;
}
```

Now we will create our custom css file for our “appId1” application:

*/client/app1/css/app1.css*

```
body{
        font-family: helvetica, arial, sans-serif;
}
```

Now to use our newly created file, we will add the following property to `bharosauio_client.properties`:

```
bharosa.uio.appId1.custom.css=/client/app1/css/app1.css
```

In this case, all we did was change helvetica to the primary font-family in our “appId1” application. Any style defined in bharosa_uio.css can be overridden in this manner if required.

## Custom Content and Messaging

Adaptive Strong Authenticator pages have a variety of content and messaging sections. These sections can be customized by properties; the default values for these are found in `bharosauio_msg.properties`. Some customizable items, like page title and message, are applicable for each page. While other items, like login blocked message, are specific to a particular page.

To change the page title on the login page in our example “appId1” application, we would add the following line to `bharosauio_client.properties`.

```
bharosa.uio.appId1.signon.page.title=Welcome to App1, please sign in.
```

Please refer to the `bharosauio_client.properties` for additional properties.

The contents of error messages are also controlled in the same way. In the following example we will customize the error message displayed when a user has been blocked by security rules.

```
bharosa.uio.appId1.login.user.blocked = You are not authorized to
login. Please contact customer service at 1-888-555-1234.
```

## How the Properties Work

An application in Adaptive Strong Authenticator is made up of a grouping or set of properties. You can configure the Adaptive Strong Authenticator properties on a global or application specific level.

The Adaptive Strong Authenticator property names are prefixed with `bharosa.uio`. They are followed by the Application ID or "default" if the setting is global.

The `bharosa.uio.default.header` property, shown below, defines the location of the header file.

```
bharosa.uio.default.header = /header.jsp
```

The property is used across all applications of the Adaptive Strong Authenticator installation unless the specific application has another location specified.

In the case shown above, "default" is used instead of the Application ID to designate the property as a global default. If the same property is not defined for an application; then, this value will be used.

If the Adaptive Strong Authenticator installation has two applications and the first application (appId1) uses the default header, but the second application (appId2) uses a different header, the following line may be added.

```
bharosa.uio.appId2.header = /client/app2/header.jsp
```

### Property Extension

In addition to configuring properties for each application, you can configure a set of properties that several applications have in common. You can then extend that set to customize the parameters that differ between the set of applications.

If you were to configure three applications that all use a single footer, but each has a unique header, you can include the following properties:

```
bharosa.uio.myAppGroup.footer = /myAppGroup/header.jsp


bharosa.uio.appId1.extends=myAppGroup
bharosa.uio.appId1.header=/client/app1/header.jsp


bharosa.uio.appId2.extends=myAppGroup
bharosa.uio.appId2.header==/client/app2/header.jsp


bharosa.uio.appId3.extends=myAppGroup
bharosa.uio.appId3.header==/client/app3/header.jsp
```

### User-Defined Enums

User-defined enums are a collection of properties that represent a list of items. Each element in the list may contain several different attributes. The definition of a user-defined enum begins with a property ending in the keyword "`.enum`" and has a value describing the use of the user-defined enum. Each element definition then starts with the same property name as the enum, and adds on an element name and has a value of a unique integer as an ID. The attributes of the element follow the same pattern, beginning with the property name of the element, followed by the attribute name, with the appropriate value for that attribute.

The following is an example of an enum defining credentials displayed on the login screen of an Adaptive Strong Authenticator implementation:

```
bharosa.uio.default.credentials.enum = Enum for Login Credentials
bharosa.uio.default.credentials.enum.companyid=0
bharosa.uio.default.credentials.enum.companyid.name=CompanyID
bharosa.uio.default.credentials.enum.companyid.description=Company ID
bharosa.uio.default.credentials.enum.companyid.inputname=comapanyid
bharosa.uio.default.credentials.enum.companyid.maxlength=24
bharosa.uio.default.credentials.enum.companyid.order=0

bharosa.uio.default.credentials.enum.username=1
bharosa.uio.default.credentials.enum.username.name=Username
bharosa.uio.default.credentials.enum.username.description=Username
bharosa.uio.default.credentials.enum.username.inputname=userid
bharosa.uio.default.credentials.enum.username.maxlength=18
bharosa.uio.default.credentials.enum.username.order=1
```

This set of properties defines one user-defined enum that contains two elements, each of which with five attributes. The “name” and “description” attributes are required to define any user-defined enum, other attributes are defined and used as needed by each individual use of a user-defined enum.

## Overriding Existing User-Defined Enums

Overriding existing user-defined enums has some special cases. You may override any existing enum element’s attribute value of the default application ID just as you would any other property, but in order to change the value of an element’s attribute in a single application using an `appId`, you must create the entire enum in that application using the appropriate `appId`.

For example, using the User Defined Enum defined above, if we wanted to change “`Company ID`” to “`Profile ID`” for only one application (`appId1`), we would need to do the following:

```
bharosa.uio.appId1.credentials.enum = Enum for Login Credentials
bharosa.uio.appId1.credentials.enum.profileid=0
bharosa.uio.appId1.credentials.enum.profileid.name=ProfileID
bharosa.uio.appId1.credentials.enum.profileid.description=Profile ID
bharosa.uio.appId1.credentials.enum.profileid.inputname=profileid
bharosa.uio.appId1.credentials.enum.profileid.maxlength=20
bharosa.uio.appId1.credentials.enum.profileid.order=0

bharosa.uio.appId1.credentials.enum.username=1
bharosa.uio.appId1.credentials.enum.username.name=Username
bharosa.uio.appId1.credentials.enum.username.description=Username
bharosa.uio.appId1.credentials.enum.username.inputname=userid
bharosa.uio.appId1.credentials.enum.username.maxlength=18
bharosa.uio.appId1.credentials.enum.username.order=1
```

### Disabling Elements

To disable any already defined element in a user-defined enum, simply add an "`enabled`" attribute with a value of "`false`". Using the `appId1` credentials enum from above, we would add the following line to remove "`Profile ID`" from the elements used by the application:

```
bharosa.uio.appId1.credentials.enum.profileid.enabled=false
```

# Authenticator Properties

Each Authenticator interface has its own unique security features. Some of these features can be enabled and disabled by editing a properties file. The following properties can be configured by adding them to `bharosauio_client.properties`.

## TextPad

The TextPad is a personalized device for entering passwords or PIN using regular keyboard. Like other Adaptive Strong Authentication devices, this also helps in solving the phishing problem.


| | |
|---|---|
| Default BG<br>(Can be application specific) | `bharosa.uio.<appId>.DeviceTextPad.default.image = textpad_bg/UIO_BG.jpg` |
| Password Frame File<br>(Can be application specific) | `bharosa.uio.<appId>.password.DeviceTextPad.frame =` |
| Challenge Frame File<br>(Can be application specific) | `bharosa.uio.<appId>.challenge.DeviceTextPad.frame =` |
| Registration Frame File<br>(Can be application specific) | `bharosa.uio.<appId>.register.DeviceTextPad.frame = textpad_bg/TP_O_preview.png` |
| User Preferences Frame File<br>(Can be application specific) | `bharosa.uio.<appId>.userpreferences.DeviceTextPad.frame = textpad_bg/TP_O_preview.png` |
| Global TextPad Input Field Maxlength | `bharosa.authentipad.textpad.datafield.maxLength=25` |
| Global TextPad Timestamp Color | `bharosa.authentipad.textpad.timestamp.font.color = ffffff` |
| Global TextPad Caption Color | `bharosa.authentipad.textpad.caption.font.color = 000000` |

## KeyPad

The KeyPad is a customizable graphics keyboard, which can be used to enter Alpha Numeric and special character that can be entered using the traditional keyboard. It is ideal for entering passwords and other sensitive data like credit card numbers, etc.


| | |
|---|---|
| Default BG<br>(Can be application specific) | `bharosa.uio.<appId>.DeviceKeyPadFull.default.image = keypad_bg/UIO_BG.jpg` |
| Password Frame File<br>(Can be application specific) | `bharosa.uio.<appId>.password.DeviceKeyPadFull.frame =` |
| Challenge Frame File<br>(Can be application specific) | `bharosa.uio.<appId>.challenge.DeviceKeyPadFull.frame =` |
| Registration Frame File<br>(Can be application specific) | `bharosa.uio.<appId>.register.DeviceKeyPadFull.frame = alphapad_bg/kp_O_preview.png` |
| User Preferences Frame File<br>(Can be application specific) | `bharosa.uio.<appId>.userpreferences.DeviceKeyPadFull .frame = alphapad_bg/kp_O_preview.png` |
| Global KeyPad Key Skins | `bharosa.authentipad.full.skins.dirlist=alphapad_skins/square` |
| Global KeyPad Timestamp Color | `bharosa.authentipad.full.timestamp.font.color = ffffff` |
| Global KeyPad Caption Color | `bharosa.authentipad.full.caption.font.color = 000000` |

## PinPad

The PinPad is a lightweight authentication device to enter numeric PIN.


| | |
|---|---|
| Default BG<br>(Can be application specific) | `bharosa.uio.default.DevicePinPad.default.image = pinpad_bg/UIO_BG.jpg` |
| Password Frame File<br>(Can be application specific) | `bharosa.uio.<appId>.password.DevicePinPad.frame =` |
| Challenge Frame File<br>(Can be application specific) | `bharosa.uio.<appId>.challenge.DevicePinPad.frame =` |
| Registration Frame File<br>(Can be application specific) | `bharosa.uio.<appId>.register.DevicePinPad.frame = pinpad_bg/PP_v02_frame_preview.png` |
| User Preferences Frame File<br>(Can be application specific) | `bharosa.uio.<appId>.userpreferences.DevicePinPad.frame = pinpad_bg/PP_v02_frame_preview.png` |
| Global PinPad Timestamp Color | `bharosa.authentipad.numeric.timestamp.font.color = ffffff` |
| Global PinPad Key Skins | `bharosa.authentipad.numeric.skins.dirlist=pinpad_skins/square,pinpad_skins/oval,pinpad_skins/hexa` |
| Global PinPad Timestamp Color | `bharosa.authentipad.numeric.timestamp.font.color = ffffff` |
| Global PinPad Caption Color | `bharosa.authentipad.numeric.caption.font.color = 000000` |

## QuestionPad

The QuestionPad is a personalized device for entering answers to challenge questions using regular keyboard. The QuestionPad is capable of incorporating the challenge question into QuesitonPad image. Like other Adaptive Strong Authentication devices, this also helps in solving the phishing problem.


| | |
|---|---|
| Default BG<br><br>(Can be application specific) | `bharosa.uio.<appId>.DeviceQuestionPad.default.image = textpad_bg/UIO_BG.jpg` |
| Challenge Frame File<br><br>(Can be application specific) | `bharosa.uio.<appId>.challenge.DeviceQuestionPad.frame =` |
| Global QuestionPad Timestamp Color | `bharosa.authentipad.questionpad.timestamp.font.color = ffffff` |
| Global QuestionPad Caption Color | `bharosa.authentipad.questionpad.caption.font.color = 000000` |