This document was last updated 2/25/11.

# Errata for Professional Oracle WebLogic Server — 9780470484302

While we made every effort to ensure the accuracy of the book, a few errors slipped through into the final printed version. This document is our attempt to document those errors for our readers. Note that this list is a list of known errors in the first printing of the book. Depending on which printing you have, some of these issues may not be present since they were addressed in later printings.

## About the Authors

- On page vii, Greg Nyberg's correct title is Senior Development Engineer, IT, at Carlson Hotels Worldwide.
- On page vii, the third sentence in Paul Done's bio has an extraneous ''the'' before the text ''Oracle's Designer 2000 product.''
- On page vii, the last sentence has an extraneous ''a'' between ''and'' and ''is.''

## Chapter 6

- On page 172, the Best Practice uses the word ''corrupt'' when ''pollute'' would have been better.
- On page 209, the second paragraph mentions that many features are only available via product-specific annotations and not available via vendor descriptors. To be clear, we should have said that they were not available via vendor-specific descriptors.

## Chapter 8

- On page 272, the fifth sentence in the second paragraph is missing the word ''at'' and should have said:

  We look at integration with remote and foreign JMS providers in Chapter 10.
- On page 281, Table 8-3 incorrectly references WLST in reference to ''WebLogic Server module configuration'' when it should have referenced WLDF instead.

- On page 292, the first sentence should have been qualified to say:

  Typically, IDEs encourage a separate project per module or library, and one for the enterprise application.

# Chapter 9

- On page 311, the last sentence of the paragraph directly below the Best Practice incorrectly states that the servlet mapping is to the `WebServiceServlet` when, in fact, it is to the JWS class. As such, the sentence should read:

  The `web.xml` deployment descriptor contains a servlet mapping to the JWS class, which WebLogic Server detects as being a Web Service and internally dispatches the SOAP requests to WebLogic Server's Web Services container.

- On page 325, the second sentence of the last paragraph mistakenly implies that WebLogic Server always ignores the `SOAPAction` header. Actually, the default behavior is to ignore the `SOAPAction` header, though this behavior can be overridden. As such, the sentence should read:

  Unfortunately, this mechanism is transport-specific and can legally be ignored or overruled by the web service implementation, as it is by default with WebLogic Server.

- On page 326, the first sentence in the second paragraph incorrectly states that only the `Action` element is mandatory when in fact both the `To` and `Action` elements are mandatory.

- On page 326, the second sentence in the second paragraph states that the `To` element identifies the URL of the target web service and then uses URL as an example. While this is not blatantly wrong, a better version of the sentence would be:

  The `To` element identifies the unique location of the target web service (for example, its URL) and the `Action` element names the target operation.

- On page 331, the use of the word ''type'' in the second sentence of the last paragraph is confusing. We should have been more explicit and said:

  If we use `javax.xml.transform.Source` instead of `javax.xml.soap.SOAPMessage` as the `invoke()` method's input parameter and return types, we can use alternative mechanisms for processing the XML data of web service requests and responses.

- On page 338, the second sentence in the last paragraph incorrectly mixes singular and plural forms. It should say:

  This is particularly useful when you need to pass a binary file, such as a JPEG image or PDF document, to a web service.

- On page 339, the line that creates the temp file in Listing 9-9 needs to be changed so that it works properly on all platforms. The new line should be:

```
File tempFile = File.createTempFile("TmpAttmnt_", ".tmp");"
```

- On page 342, the first sentence of the second paragraph in the ''Implementing Stateful Web Services'' section should read:

  If we want to ignore this best practice, WebLogic Server provides a mechanisms that allows us to maintain session state between web service requests.

- On page 351, the first sentence in the number 1 element of the numbered list should say:

  Further configure the SSL settings for each server in the domain.

- On page 352, the SOAP Message example in the ''Message-level Security'' section has extraneous characters in the first line. It should read:

```
<Envelope xmlns=http://schemas.xmlsoap.org/soap/envelope/>
```

- On page 353, the second sentence of the paragraph immediately preceding Listing 9-16 should be further qualified to say:

  WebLogic Server does not provide a predefined policy file that uses a username token with a plain text password.

- On pages 354-355, the first item in the numbered list should be further qualified to say:

  Configure the WebLogic Server domain to support SSL, listening on an appropriate port (for example, 7002). Although we won't need to use HTTPS-based transport-level security in our example, we still need to make these configuration changes to enable the server's private key store and trust store to be used for message-level security.

- On page 358, the third element in bulleted list in the ''Web Services Security Configuration'' section incorrectly mixes forms by using the word ''Specify'' instead of ''Specifying.''

- On page 358, the second sentence of the second paragraph of the ''Web Services Security Configuration'' section incorrectly uses the plural form of `Web Services Security Configuration`.

- On page 359, the first sentence of the ''Adding Web Services to bigrez.com'' section incorrectly states that the PropertyServices EJB was from Chapter 6 when it actually is discussed in Chapter 7.

# Chapter 10

- On pages 366-367, the second sentence in the first paragraph below the list of acknowledgement modes starts with the words ''in contrast'' but would be clearer if it started with ''however'' instead.

- On page 381, the last sentence in the description of the Java Install Client would be clearer if we replace the phrase ''when calling into JMS APIs'' with ''when using JMS APIs.''

- On page 391, the first paragraph mentions that JDBC stores may provide an easier solution for addressing failover issues because the database typically resides on another machine. While this is part of the reason, the other part that we failed to mention is that many times the database will already have some sort of high availability solution in place to prevent the database machine itself from becoming a single point of failure.

- On page 407, the third sentence in the first paragraph misspells the word ''predictable.''

- On page 429, the third sentence in the first paragraph incorrectly mixes verb tenses so the word ''start'' should be replaced by ''starting'' and the word ''use'' should be replaced by ''using.''

- On page 433, we refer to WebLogic JMS's Foreign Server support as new. While it is new since the first edition of this book was published in 2003, it is not new in the 11g release.

# Chapter 11

- On page 458, the last paragraph refers to the ''Setting Up WebLogic Server Application Security'' section later in this chapter.
- On page 463, the last sentence in the ''Exporting Passwords'' list item should be further clarified to say that enabling this option requires you to reset all existing user passwords so that the `DefaultAuthenticator` can store them using a reversible encryption scheme.
- On page 483, the second sentence in the first paragraph is repeated from the previous page and should have been removed.
- On pages 490-491, we discuss three features that rely on the SSL transport to work:
  - `<transport-guarantee>`
  - `<cookie-secure>`
  - AuthCookie

  While the discussion is accurate for WLS 10.3 and 10.3.1, there was a bug (Bug 8254839) that made the product not support these three features when SSL was terminated at the WebLogic Server web server plug-in or a hardware load balancer. WebLogic Server 10.3.2 includes the fix for this bug.

  By design, these three features should work because the plug-in adds a `WL-Proxy-SSL` HTTP header to the request when the web server receives a request over SSL. When a WebLogic Server instance receives a request with `WL-Proxy-SSL` set to `true`, it checks to see if its `WebLogic Plug-In Enabled` attribute is set. If so, it treats the request as coming over SSL even though the request was transmitted over clear text between the plug-in and the WebLogic Server instance. Set this attribute in the `Advanced` section of the cluster's `General Configuration` tab if using a cluster and in the `Advanced` section of the server's `General Configuration` tab if not using a cluster.

  Some hardware load balancers also have the ability to add the `WL-Proxy-SSL` header for requests being sent to WebLogic Server. You should note that enabling the `WebLogic Plug-In Enabled` attribute makes it possible for rogue clients to set the `WL-Proxy-SSL` header and gain access to these secure features designed to work over SSL without requiring the clients to actually use SSL. As such, you probably want to make sure that any servers that set the `WebLogic Plug-In Enabled` attribute are not directly accessible and only allow access through a web server using the WebLogic Server plug-in or a hardware load balancer.

  If using a hardware load balancer directly against WebLogic Server instances or clusters, you should configure the load balancer to strip off any `WL-Proxy-SSL` header it finds on incoming request to block rogue clients from sending this header through the load balancer to the server. You may also want to configure the load balancer to strip off any WebLogic Server-specific headers from the responses so as not to expose internal information the server returns for the plug-in's benefit to clients. When using the WebLogic Server web server plug-ins, this step is not needed since the plug-in will do that for you.

  One additional point to mention is that, by default, WebLogic Server does not allow applications to access the plug-in's HTTP headers. Since these headers are meant solely to help the plug-in and the server work together, applications usually do not need to concern themselves with these headers. If you find yourself needing access to these headers from within your application, you need to set the Java system property `weblogic.http.isWLProxyHeadersAccessible` to `true` on the command line used to start WebLogic Server.

- On page 499, we refer to the Advanced mode but probably do not give sufficient context for the reader to understand that this is one of the four authorization models that Weblogic Server's default authorization provider supports that we discussed back on page 454.
- On page 511, the name of the Kerberos configuration files is misspelled. The corrected sentence should read:

  On Windows, create a file called `%windir%\krb5.ini;` on UNIX, create a file called `/etc/krb5.conf.`
- On page 517, we should have pointed out that a servlet authentication filter is only invoked when an unauthenticated client requests access to a protected resource, which is the same behavior that all other web application authentication mechanisms exhibit.