Oracle® Coherence
Release Notes for Oracle Coherence
Release 3.5.2
E15884-01

October 2009
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preface</strong></td>
<td>vii</td>
</tr>
<tr>
<td>Audience</td>
<td>vii</td>
</tr>
<tr>
<td>Documentation Accessibility</td>
<td>vii</td>
</tr>
<tr>
<td>Related Documents</td>
<td>viii</td>
</tr>
<tr>
<td>Conventions</td>
<td>viii</td>
</tr>
</tbody>
</table>

## 1 Technical Changes and Enhancements

**Oracle Coherence for Java 3.5.2** ................................................................. 1-1

- Coherence*Web Enhancements and Fixes ...................................................... 1-1
- Management Framework Fixes ....................................................................... 1-1
- Other Enhancements and Fixes .................................................................... 1-1

**Oracle Coherence for .NET 3.5.2** ............................................................. 1-2

**Oracle Coherence for C++ 3.5.2** .............................................................. 1-2

## 2 Documentation Errata

- Changes to Defining a Data Grid............................................................... 2-1
- Changes to Production Checklist .............................................................. 2-2
- Changes to Deliver Events for Changes as they Occur ............................. 2-2
- Changes to Operational Configuration Deployment Descriptor .................. 2-2
- Changes to Links in the bdb-store-manager Element ................................ 2-3
- Changes to the member-identity Element .................................................. 2-3
- Changes to the distributed-scheme Element .............................................. 2-3
- Changes to the tcp-acceptor Element ...................................................... 2-4
- Additions to the Log Message Glossary .................................................. 2-4

**Using PofExtractors and PofUpdaters** ..................................................... 2-5

- Navigating a POF object ............................................................................ 2-6
- Using PofExtractors .................................................................................. 2-7
- Using PofUpdaters .................................................................................... 2-8

**Index**
List of Examples

2–1    A Java Class with Indexed Data Members ................................................................. 2-6
### List of Tables

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2–1</td>
<td>Changes to Defining a Data Grid</td>
<td>2-1</td>
</tr>
<tr>
<td>2–2</td>
<td>Changes to Production Checklist Description</td>
<td>2-2</td>
</tr>
<tr>
<td>2–3</td>
<td>Changes to Linux Support</td>
<td>2-2</td>
</tr>
<tr>
<td>2–4</td>
<td>Changes to member-identity Element Description</td>
<td>2-3</td>
</tr>
<tr>
<td>2–5</td>
<td>Defaults for Example Service Storage Sizes</td>
<td>2-4</td>
</tr>
</tbody>
</table>
Preface

This document describes changes and enhancements that have been made to the Oracle Coherence product since the 3.5.1 release.

Audience

This document is intended for users of Oracle Coherence.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at http://www.fcc.gov/cgb/consumerfacts/trs.html, and a list of phone numbers is available at http://www.fcc.gov/cgb/dro/trsphonebk.html.
Related Documents

For more information, see the following documents in the Oracle Coherence documentation set:

■ Getting Started with Oracle Coherence
■ Developer’s Guide for Oracle Coherence
■ Client Guide for Oracle Coherence
■ Tutorial for Oracle Coherence
■ User’s Guide for Oracle Coherence*Web
■ Integration Guide for Oracle Coherence

Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>boldface</td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td>italic</td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td>monospace</td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
This chapter describes the changes and enhancements made to the Oracle Coherence product for the 3.5.2 release. This document is accurate at the time of publication. Oracle updates the release notes periodically after the software release.

Oracle Coherence for Java 3.5.2

The following is a list of new features, improvements, and bug fixes in Oracle Coherence for Java 3.5.2:

Coherence*Web Enhancements and Fixes

- ClassNotFoundException no longer occurs if multiple web applications are deployed using Application or EAR scoped cluster nodes.
- Enhanced Coherence*Web to allow custom attribute serialization.
- Enhanced Coherence*Web SPI to use new Coherence and Coherence*Web features in WebLogic 10.3.3.
- Fixed the optimistic locking mode to ensure that updates to sessions that have been concurrently invalidated are rejected.

Management Framework Fixes

- Reporter now properly starts when Coherence is deployed as a shared library, for example, with Coherence*Web.
- Fixed a ClassCastException which could occur when MBeans are removed while Reporter is in the middle of executing a report.

Other Enhancements and Fixes

- Enhanced PofExtractor to allow clear indication of expected data type, to ensure that proper results are being returned for certain data types where ambiguity may lead to incorrect results being returned, such as intrinsic numeric data types, primitive arrays, and so on.
- Fixed regression in CacheFactory and DefaultConfigurableCacheFactory behavior.
- Fixed regression in ContinuousQueryCache.
- Improved clarity of Guardian logs on daemon-pool worker thread recovery/termination.
• Fixed NPE in Daemon.isGuarded().
• Fixed infinite loop in entrySet call during eviction.
• Optimized Entry initialization for custom backing maps.
• Added hashcode and equals implementation for NotFilter.
• Optimized off-heap partition initialization.
• Fixed ArrayIndexOutOfBoundsException in SafeSortedMap.
• Enhanced the clarity of configuration files by allowing unit-factor to be specified in units, that is, K, M, G.

Oracle Coherence for .NET 3.5.2

The following is a list of new features, improvements, and bug fixes in Oracle Coherence for .NET 3.5.2:

• Optimized Filter-based cache events to avoid unnecessary reevaluation on the client.
• Enhanced PofExtractor to allow clear indication of expected data type, to ensure that proper results are being returned for certain data types where ambiguity may lead to incorrect results being returned, such as intrinsic numeric data types, primitive arrays, and so on.
• Fixed regression in ContinuousQueryCache.release() which may have resulted in IllegalStateException.

Oracle Coherence for C++ 3.5.2

The following is a list of new features, improvements, and bug fixes in Oracle Coherence for C++ 3.5.2:

• Enhanced PofExtractor to allow clear indication of expected data type, to ensure that proper results are being returned for certain data types where ambiguity may lead to incorrect results being returned, such as intrinsic numeric data types, primitive arrays, and so on.
• Made HashSet's constructors private, to prevent illegal stack based allocation.
• Fixed regression in ContinuousQueryCache.release() which may have resulted in IllegalStateException.
This chapter describes changes, enhancements, and corrections made to the Oracle Coherence documentation library for the 3.5.2 release. The library can be found at the following URL:

http://download.oracle.com/docs/cd/E14526_01/index.htm

This chapter contains the following sections:

- Changes to Defining a Data Grid
- Changes to Production Checklist
- Changes to Deliver Events for Changes as they Occur
- Changes to Operational Configuration Deployment Descriptor
- Changes to Links in the bdb-store-manager Element
- Changes to the member-identity Element
- Changes to the distributed-scheme Element
- Changes to the tcp-acceptor Element
- Additions to the Log Message Glossary
- Using PofExtractors and PofUpdaters

Changes to Defining a Data Grid

A change has been made to the Defining a Data Grid chapter of Getting Started with Oracle Coherence. The reference to Ruby as a supported object-oriented language for an application object has been removed.

<table>
<thead>
<tr>
<th>Old Text</th>
<th>New Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>The application objects are the actual components of the application that contain the information shared across multiple servers. These objects must survive a possible server failure in order for the application to be continuously available. These objects are typically built in an object-oriented language such as Java (e.g. POJOs), C++, C#, VB.NET or Ruby. Unlike a relational schema, the application objects are often hierarchical and may contain information that is pulled from any database.</td>
<td>The application objects are the actual components of the application that contain the information shared across multiple servers. These objects must survive a possible server failure in order for the application to be continuously available. These objects are typically built in an object-oriented language such as Java (for example, POJOs), C++, C#, or VB.NET. Unlike a relational schema, the application objects are often hierarchical and may contain information that is pulled from any database.</td>
</tr>
</tbody>
</table>
Changes to Production Checklist

A addition has been made to the Large Cluster Configuration section of the Production Checklist appendix in the Developer’s Guide for Oracle Coherence.

<table>
<thead>
<tr>
<th>Table 2–2 Changes to Production Checklist Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Old Text</strong></td>
</tr>
<tr>
<td>Coherence clusters which consist of over 400 TCMP</td>
</tr>
<tr>
<td>nodes need to increase the default maximum packet</td>
</tr>
<tr>
<td>size Coherence will utilize. The default of 1468</td>
</tr>
<tr>
<td>should be increased relative to the size of the</td>
</tr>
<tr>
<td>cluster, i.e. a 600 node cluster would need the</td>
</tr>
<tr>
<td>maximum packet size increased by 50%. The maximum</td>
</tr>
<tr>
<td>packet size is configured as part of the coherence</td>
</tr>
<tr>
<td>operational configuration file, see [packet-size]</td>
</tr>
<tr>
<td>for details on changing this setting.</td>
</tr>
</tbody>
</table>

A change has been made to the question What are the supported and suggested server operating systems for deploying Coherence on? in the Operating Systems section of the Production Checklist appendix.

<table>
<thead>
<tr>
<th>Table 2–3 Changes to Linux Support</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Old Text</strong></td>
</tr>
<tr>
<td>For commodity x86 servers, Linux</td>
</tr>
<tr>
<td>distributions based on the Linux</td>
</tr>
<tr>
<td>2.6 kernel are recommended. While</td>
</tr>
<tr>
<td>it is expected that most 2.6-based</td>
</tr>
<tr>
<td>Linux distributions will provide a</td>
</tr>
<tr>
<td>good environment for running</td>
</tr>
<tr>
<td>Coherence, the following are</td>
</tr>
<tr>
<td>recommended by Oracle: RedHat</td>
</tr>
<tr>
<td>Enterprise Linux (version 4 or</td>
</tr>
<tr>
<td>later) and Suse Linux Enterprise</td>
</tr>
<tr>
<td>(version 10 or later). Oracle also</td>
</tr>
<tr>
<td>routinely tests using</td>
</tr>
<tr>
<td>distributions such as</td>
</tr>
<tr>
<td>RedHat Fedora Core 5 and even</td>
</tr>
<tr>
<td>Knoppix “Live CD”.</td>
</tr>
</tbody>
</table>

Changes to Deliver Events for Changes as they Occur

In the Advanced: Listening to Queries section of Deliver Events for Changes as they Occur, in Getting Started with Oracle Coherence, the name of the trades class (for example, trades.addMapListener(listener, new MapEventFilter(filter), true)) has been changed to mapTrades (for example, mapTrades.addMapListener(listener, new MapEventFilter(filter), true)).

Changes to Operational Configuration Deployment Descriptor

A change has been made to the DOCTYPE declaration for the Coherence Operational Configuration deployment descriptor in the Document Format section of the Operational Configuration Elements appendix of the Developer’s Guide for Oracle Coherence. The DOCTYPE declaration has been changed from this:

```xml
<!DOCTYPE coherence PUBLIC "-//Oracle, Inc.//DTD Oracle Coherence 3.5//EN" "http://www.tangosol.com/dtd/coherence_3_3.dtd">
```

To this:

```xml
<!DOCTYPE coherence PUBLIC "-//Oracle, Inc.//DTD Oracle Coherence 3.5//EN" "http://www.tangosol.com/dtd/coherence_3_3.dtd">
```
Changes to the distributed-scheme Element

Changes to the member-identity Element

Changes have been made to the descriptions of the following member-identity sub-elements in the Operational Configuration Elements appendix in the Developer’s Guide for Oracle Coherence:

<table>
<thead>
<tr>
<th>Table 2–4</th>
<th>Changes to member-identity Element Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old Text</td>
<td>New Text</td>
</tr>
<tr>
<td>&lt;site-name&gt; (Optional) The site-name element contains the name of the geographic site that the member is hosted at. For WAN clustering, this value identifies the datacenter within which the member is located, and can be used as the basis for intelligent routing, load balancing and disaster recovery planning (i.e. the explicit backing up of data on separate geographic sites). The name is also useful for displaying management information (e.g. JMX) and interpreting log entries. It is optional to provide a value for this element. Deployments that spread across more than one geographic site should specify a site-name value.</td>
<td>&lt;site-name&gt; (Optional) The site-name element contains the name of the geographic site that the member is hosted at. For WAN clustering, this value identifies the datacenter within which the member is located, and can be used as the basis for intelligent routing, load balancing and disaster recovery planning (i.e. the explicit backing up of data on separate geographic sites). The name is also useful for displaying management information (e.g. JMX) and interpreting log entries. Coherence currently does not make use of this element in making decisions about data backup location. It is optional to provide a value for this element.</td>
</tr>
<tr>
<td>&lt;rack-name&gt; (Optional) The rack-name element contains the name of the location within a geographic site that the member is hosted at. This is often a cage, rack or bladeframe identifier, and can be used as the basis for intelligent routing, load balancing and disaster recovery planning (i.e. the explicit backing up of data on separate bladeframes). The name is also useful for displaying management information (e.g. JMX) and interpreting log entries. It is optional to provide a value for this element. Large scale deployments should always specify a rack-name value.</td>
<td>&lt;rack-name&gt; (Optional) The rack-name element contains the name of the location within a geographic site that the member is hosted at. This is often a cage, rack or bladeframe identifier, and can be used as the basis for intelligent routing, load balancing and disaster recovery planning (i.e. the explicit backing up of data on separate bladeframes). The name is also useful for displaying management information (for example, JMX) and interpreting log entries. Coherence currently does not make use of this element in making decisions about data backup location. It is optional to provide a value for this element.</td>
</tr>
</tbody>
</table>

Changes to the distributed-scheme Element

Changes have been made to the descriptions of the partitioned and partition-count sub-elements of the distributed-scheme element in the Cache Configuration Elements appendix in the Developer’s Guide for Oracle Coherence.

The following changes were made to the note in the partitioned description:
When using an off-heap backing map it is important that the corresponding backup-storage be configured for off-heap (potentially using the same scheme as the backing-map). Here off-heap refers to any storage where some or all entries are stored outside of the JVMs garbage collected heap space. Examples include overflow-scheme, and external-scheme. See the Partitioned cache with overflow sample in the Sample Cache Configurations appendix for an example configuration.

The following is the new description of the partitioned-count sub-element:

\[<\text{partition-count}>\] (Optional)—Specifies the number of partitions that a partitioned (distributed) cache will be "chopped up" into. Each member running the partitioned cache service that has the local-storage option set to true will manage a "fair" (balanced) number of partitions.

Legal values are positive integers. The number of partitions should be a prime number and sufficiently large such that a given partition is expected to be no larger than 50MB. A list of first 1,000 primes can be found at this URL: http://www.utm.edu/research/primes/lists/small/1000.txt

Good default values for example service storage sizes are provided in Table 2–5:

<table>
<thead>
<tr>
<th>Service Storage Size</th>
<th>partition-count Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>100M</td>
<td>257</td>
</tr>
<tr>
<td>1G</td>
<td>509</td>
</tr>
<tr>
<td>10G</td>
<td>2039</td>
</tr>
<tr>
<td>50G</td>
<td>4093</td>
</tr>
<tr>
<td>100G</td>
<td>8191</td>
</tr>
</tbody>
</table>

### Changes to the tcp-acceptor Element

A change has been made to the description of the tcp-acceptor element in the Cache Configuration Elements appendix in the Developer’s Guide for Oracle Coherence.

The \(<\text{tcp-acceptor}>\) element can use either an \(<\text{address-provider}>\) or a \(<\text{local-address}>\) element.

The address-provider sub-element specifies the configuration for the com.tangosol.util.AddressProvider address factory that will supply the local address (IP or DNS name) and port that the TCP/IP ServerSocket opened by the connection acceptor will listen on.

### Additions to the Log Message Glossary

The following error messages have been added to the Log Message Glossary appendix in the Developer’s Guide for Oracle Coherence.

**Failover Related Messages**

validatePolls: This service timed-out due to unanswered handshake request. Manual intervention is required to stop the members that have not responded to this Poll.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>none</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity</td>
<td>1-Error</td>
</tr>
</tbody>
</table>
Using PofExtractors and PofUpdaters

In Coherence, extractors and updaters are used to extract and update values of objects that are stored in the cache. PofExtractors and PofUpdaters take advantage of POFs indexed state to extract or update an object without requiring full serialization or deserialization routines.

PofExtractors and PofUpdaters adds flexibility in working with non-primitive types in Coherence. For most cases, where you’re working with extend clients, it is no longer required to have corresponding Java classes in the grid. Because PofExtractors and PofUpdaters can navigate the binary, the entire key/value
does not have to be deserialized into Object form. This implies that indexing can be achieved by simply using a PofExtractor to pull a value to index on. There are, however, circumstances where you must provide a corresponding Java class:

- **Key Association**—When using key association, Coherence always deserializes keys to determine whether they implement KeyAssociation.

- **Cache Stores**—When using a cache store, Coherence passes the deserialized version of the key and value to the cache store to write to the back end.

### Navigating a POF object

Due to the fact that the Portable Object Format is indexed, it is possible to quickly traverse the binary to a specific element for extraction or updating. It is the responsibility of the com.tangosol.io.pof.reflect.PofNavigator interface to traverse a POF value object and return the desired POF value object. Out of the box, Coherence provides a com.tangosol.io.pof.reflect.SimplePofPath class that can navigate a POF value based on integer indexes. In the simplest form, all you must do is to provide the index of the attribute to extract or update.

The Contact class illustrated in Example 2–1 defines a constant for each data member that is being written to and from the POF stream. This is an excellent practice to follow as it will simplify both writing your serialization routines as well as making it easier to work with PfoExtractors and PfoUpdaters. By labeling each index, it becomes much easier to think about what you are working with. As mentioned earlier, in the simplest case, you could pull the work address out of the contact by using the WORK_ADDRESS index. The SimplePofPath also allows using an Array of ints to traverse the PofValues. For example, if you wanted the zip code of the work address you would use [WORK_ADDRESS, ZIP]. The following sections will go through the example in greater detail.

#### Example 2–1  A Java Class with Indexed Data Members

```java
public class Contact
    implements PortableObject
{
    ...
    // ----- PortableObject interface --------------------------------------
    /** *
     * @inheritDoc
     */
    public void readExternal(PofReader reader)
        throws IOException
    {
        m_sFirstName     = reader.readString(FIRSTNAME);
        m_sLastName      = reader.readString(LASTNAME);
        m_addrHome       = (Address) reader.readObject(HOME_ADDRESS);
        m_addrWork       = (Address) reader.readObject(WORK_ADDRESS);
        m_mapPhoneNumber = reader.readMap(PHONE_NUMBERS, null);
    }

    /** *
     * @inheritDoc
     */
    public void writeExternal(PofWriter writer)
        throws IOException
    {
        writer.writeString(FIRSTNAME, m_sFirstName);
        writer.writeString(LASTNAME, m_sLastName);
    }
```

2-6  Oracle Coherence Release Notes for Oracle Coherence
Using PofExtractors and PofUpdaters

Using PofExtractors

Extractors are typically used when querying the cache. Using a PofExtractor should greatly improve the performance of your queries. If you were to use the class illustrated in Example 2-1, and wanted to query the cache for all Contacts with the last names Jones, the query would look something like this:

```java
ValueExtractor veName = new PofExtractor(String.class, Contact.LASTNAME);
Filter filter = new EqualsFilter(veName, "Jones");
Set setEntries = cache.entrySet(filter);
```

In this use case, the PofExtractor has a convenience constructor that will use a SimplePofPath to retrieve a singular index, in this case, the Contact.LASTNAME index. Now, if you wanted to find all Contacts with the area code 01803, the query would look like this:

```java
ValueExtractor veZip = new PofExtractor(
```
Using PofExtractors and PofUpdaters

```java
String.class, new SimplePofPath(new int[]{Contact.WORK_ADDRESS, Address.ZIP});
Filter filter = new EqualsFilter(veZip, "01803");

// find all entries that have a work address in the 01803 zip code
Set setEntries = cache.entrySet(filter);
```

Notice that in the previous examples, the PofExtractor constructor has a first argument (added in 3.5.2) with the class of the extracted value or null. The reason for passing type information is that POF uses a compact form in the serialized value when possible. For example, some numeric values are represented as special POF intrinsic types in which the type implies the value. As a result, POF requires the receiver of a value to have implicit knowledge of the type. PofExtractor uses the class supplied in the constructor as the source of the type information. If the class is null, PofExtractor will infer the type from the serialized state, but the extracted type may differ from the expected type. Strings, in fact, can be correctly inferred from the POF stream, so null is sufficient in the previous examples. In general, however, null should not be used.

**Using PofUpdaters**

PofUpdaters work in the same way as PofExtractors except that they update the value of an object rather than extract it. So, to change all entries with the last name of Jones to Smith, use the UpdaterProcessor like this:

```java
ValueExtractor veName = new PofExtractor(String.class, Contact.LASTNAME);
Filter filter = new EqualsFilter(veName, "Jones");
ValueUpdater updater = new PofUpdator(Contact.LASTNAME);

// find all Contacts with the last name Jones and change them to have the last name "Smith"
cache.invokeAll(filter, new UpdaterProcessor(updater, "Smith"));
```

**Note:** While these examples operate on String based values, this functionality will work on any POF encoded value.
Index

A
address-provider element, 2-4
ArrayIndexOutOfBoundsException, 1-2

B
bdb-store-manager element, 2-3

C
CacheFactory, 1-1
ClassCastException, 1-1
ClassNotFoundException, 1-1
Coherence clusters, default, 2-2
Coherence*Web features in WebLogic 10.3.3, 1-1
ContinuousQueryCache, 1-1
ContinuousQueryCache.release(), 1-2
custom attribute serialization, 1-1

D
Daemon.isGuarded(), 1-2
DefaultConfigurableCacheFactory, 1-1
distributed-scheme element, 2-3
DOCTYPE declaration, 2-2

E
Entry initialization, 1-2
entrySet calls, 1-2

G
Guardian logs, 1-1

H
HashSet, 1-2

I
illegal stack based allocation, 1-2
IllegalStateException, 1-2

K
KeyAssociation, 2-6

L
languages, supported, 2-1
log messages, 2-4

N
NotFilter, 1-2

O
off-heap partition initialization, 1-2
operating systems, supported, 2-2
optimistic locking mode, 1-1

P
partition-count element, 2-4
partitioned element, 2-3
PofExtractor, 1-1, 1-2, 2-5
PofUpdater, 2-5
Production Checklist, 2-2

R
rack-name element, 2-3
Reporter utility, 1-1

S
SafeSortedMap, 1-2
SimplePofPath, 2-6, 2-7
site-name element, 2-3

T
tcp-acceptor element, 2-4

U
unit-factors in configuration files, 1-2
UpdaterProcessor, 2-8