

Oracle® Application Server InterConnect

Adapter Publishing Engine User's Guide

10g (9.0.4)

Part No. B10421-01

September 2003

Oracle Application Server InterConnect Adapter Publishing Engine User's Guide, 10g (9.0.4)

Part No. B10421-01

Copyright © 2002, 2003 Oracle Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle*MetaLink*, Oracle Store, Oracle9i, Oracle9iAS Discoverer, SQL*Plus, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

| | |
|---|------------|
| Send Us Your Comments | v |
| Preface..... | vii |
| 1 Publishing Engine | |
| What is the Publishing Engine?..... | 2 |
| Supported Platforms | 3 |
| Key Management..... | 4 |
| Internal Key Management..... | 5 |
| External Key Management..... | 6 |
| Using the Publishing Engine with Oracle Application Server InterConnect..... | 7 |
| Defining Polled Events using Publishing Manager..... | 8 |
| Defining a Published Event in iStudio | 10 |
| Runtime Polling and Event Publishing using Publishing Engine..... | 11 |
| Publishing Manager Commands..... | 12 |

Index

Send Us Your Comments

Oracle Application Server InterConnect Adapter Publishing Engine User's Guide, 10g (9.0.4)
Part No. B10421-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: appserverdocs_us@oracle.com
- FAX: 650-506-7375 Attn: Oracle Application Server Documentation Manager
- Postal service:
Oracle Corporation
Oracle Application Server Documentation Manager
500 Oracle Parkway, M/S 10p6
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This preface contains these topics:

- [Intended Audience](#)
- [Documentation Accessibility](#)
- [Related Documentation](#)
- [Conventions](#)

Intended Audience

This guide is intended for those who perform the following tasks:

- install applications
- maintain applications

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. 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. Standards will continue to evolve over time, and Oracle Corporation 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 additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, 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, JAWS 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 Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Related Documentation

For more information, see these Oracle resources:

- *Oracle Application Server InterConnect User's Guide*
- *Oracle Application Server InterConnect Installation Guide*
- *Oracle Application Server InterConnect Adapter Configuration Editor User's Guide*

Printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://otn.oracle.com/membership/>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://otn.oracle.com/documentation/>

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)
- [Conventions for Microsoft Windows Operating Systems](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

| Convention | Meaning | Example |
|---|---|---|
| Bold | Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both. | When you specify this clause, you create an index-organized table . |
| <i>Italics</i> | Italic typeface indicates book titles or emphasis. | <i>Oracle9i Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk. |
| UPPERCASE monospace (fixed-width) font | Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles. | You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure. |
| lowercase monospace (fixed-width) font | Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown. | Enter sqlplus to open SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepUtil class implements these methods. |
| lowercase italic monospace (fixed-width) font | Lowercase italic monospace font represents placeholders or variables. | You can specify the <i>parallel_clause</i> . Run <code>Uold_release.SQL</code> where <i>old_release</i> refers to the release you installed prior to upgrading. |

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

| Convention | Meaning | Example |
|----------------|---|---|
| [] | Brackets enclose one or more optional items. Do not enter the brackets. | DECIMAL (<i>digits</i> [, <i>precision</i>]) |
| { } | Braces enclose two or more items, one of which is required. Do not enter the braces. | {ENABLE DISABLE} |
| | A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar. | {ENABLE DISABLE} [COMPRESS NOCOMPRESS] |
| ... | Horizontal ellipsis points indicate either: <ul style="list-style-type: none">■ That we have omitted parts of the code that are not directly related to the example■ That you can repeat a portion of the code | CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees; |
| | Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example. | |
| Other notation | You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown. | acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3; |
| <i>Italics</i> | Italicized text indicates placeholders or variables for which you must supply particular values. | CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i> |

| Convention | Meaning | Example |
|------------|--|--|
| UPPERCASE | Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase. | <pre>SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;</pre> |
| lowercase | <p>Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.</p> <p>Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.</p> | <pre>SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MUJ9;</pre> |

Conventions for Microsoft Windows Operating Systems

The following table describes conventions for Microsoft Windows operating systems and provides examples of their use.

| Convention | Meaning | Example |
|--------------------------|--|---|
| Choose Start > | How to start a program. | To start the Oracle Database Configuration Assistant, choose Start > Programs > Oracle - <i>HOME_NAME</i> > Configuration and Migration Tools > Database Configuration Assistant. |
| File and directory names | File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention. | c:\winnt "\system32 is the same as C:\WINNT\SYSTEM32 |
| C:\> | Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the <i>command prompt</i> in this manual. The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters. | C:\oracle\oradata> C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\" C:\>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept) |
| <i>HOME_NAME</i> | Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore. | C:\> net start Oracle <i>HOME_NAME</i> TNSListener |

| Convention | Meaning | Example |
|---|--|--|
| <i>ORACLE_HOME</i> and <i>ORACLE_BASE</i> | <p>In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default used one of the following names:</p> <ul style="list-style-type: none"> ■ C:\orant for Windows NT ■ C:\orawin95 for Windows 95 ■ C:\orawin98 for Windows 98 <p>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is C:\oracle. If you install Oracle9i release 1 (9.0.1) on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is C:\oracle\ora90. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p> | Go to the <i>ORACLE_BASE\ORACLE_HOME\rdms\admin</i> directory. |

Publishing Engine

This chapter provides an introduction to the Oracle Application Server InterConnect Adapter Publishing Engine. The following topics are discussed:

- [What is the Publishing Engine?](#)
- [Key Management](#)
- [Using the Publishing Engine with Oracle Application Server InterConnect](#)

What is the Publishing Engine?

The Publishing Engine provides an event publication mechanism for systems that have no native publication mechanism (for example, PeopleSoft). The Publishing Engine tracks changes to the applications by periodically polling for changes to data from operator input. It essentially polls the application looking for predefined variables.

The Publishing Engine consists of three major components:

- Publishing Engine Adapter—Provides persistent storage of event definitions and the event browsing and selection capabilities.
- Publishing Agent—Responsible for the runtime polling and event publishing. It is started as part of the Oracle Application Server InterConnect application startup.
- Publishing Manager (pubmgr)—A command line tool used to administer and configure the Publishing Engine. The Publishing Manager can configure the Publishing Engine to poll for (and publish) any information retrieved from a system.

The events defined through the Publishing Manager appears in the component browser. You can browse and define published events in iStudio.

Supported Platforms

Table 1-1 lists the platforms on which the Publishing Engine runs.

Table 1–1 Supported Platforms

| Operating System | Version |
|-------------------------|--|
| Windows NT | Version 4.0 with Service Pack 6 or above |
| Windows 2000 | With Service Pack 1 or above |
| HP-UX | 11.0 and 11.11 |
| Sun SPARC Solaris | 8 and 9 |

Key Management

To convert inbound polling calls into outbound published events, the Publishing Engine must be able to retrieve unpublished data. This is done by tracking (in persistent storage) the key fields of previously retrieved data. When the polling interval begins, the agent retrieves all data added/ changed/ deleted after the last stored key fields. After retrieving new data, the Publishing Engine publishes the data as an event and updates the stored key field. However, if no new data is retrieved, no event is generated. This cycle begins again during the next polling interval.

This process of storing and tracking key field values to retrieve only the data changed since the last polling interval is called Key Management. There are two main forms of key management recognized by the Publishing Engine:

- [Internal Key Management](#)
- [External Key Management](#)

Internal Key Management

Internal key management is where functionality, within the system itself, tracks the changes. With internal key management, the Publishing Engine does not need to store or manage keys itself. Instead, a business API is provided in the system that provides the user with all the new records since the last time this API was called. When the Publishing Engine polls this API, it may or may not retrieve records in the output set. If it retrieves records, it publishes them as a single event. Otherwise, it waits for the next polling interval.

To use internal key management with the Publishing Engine, the business API must either have no input parameters or one fixed- value input parameter. All other parameters must be output parameters (or `input/output` parameters for which the input can be empty). A fixed value input parameter is provided if the API supports multiple different callers. Each caller has a different value and the API stores caller-specific data previously retrieved.

The following is an example of an API that the Publishing Engine can support for internal key management:

```
void GetUpdatedVendors (in String callerName,  
                       out VendorList updatedVendors);
```

There is one disadvantage to use internal key management: unless the system being called supports transactional calls, the Publishing Engine can only guarantee zero- or one-time publishing of events (it cannot guarantee exactly-one time publishing of events). The API call may succeed in the system itself (and the API updates and commits its internal key storage) but if, as the data is being transmitted back to the Publishing Engine, a network (or other) failure occurs, then the Publishing Engine will not receive the data and, therefore, cannot publish it. In order to provide exactly-one time event publishing external key management must be used.

External Key Management

External key management is similar to internal key management except that the Publishing Engine manages the persistent storage of the published key values. With external key management, the business APIs used by the Publishing Engine essentially retrieves all records whose key is greater than a given input key value. The Publishing Engine stores these key values, therefore, it can provide *exactly one time* event publishing in concert with the event queuing mechanism.

To use external key management with the Publishing Engine, the business API must have one input parameter used to pass in the last known key value. This parameter can also be an input/output (the output is ignored unless the output value is the updated, most recent key value). All other parameters must be output parameters (or input/output parameters for which the input can be empty).

In addition, the API provides two ways to retrieve the most recent key value:

- There must be an output parameter which is the most recent key value.
- There is a table output parameter, which contains a field for the key value where the table is sorted so that the last row in the table is the most recent key value.

The following is an example of an API that the Event Editor can support for external key management:

```
void GetVendors (in String lastKnownKey,  
                out String mostrecentKey  
                out VendorList vendorsInAnyOrder);
```

In these cases, the Publishing Engine persistently stores and tracks the last known key value and passes the most recent value into the API each time the API is called. The Publishing Engine stores the key values (as well as the event definitions) on a per-profile basis.

Using the Publishing Engine with Oracle Application Server InterConnect

To use publishing engine with Oracle Application Server InterConnect, you need to:

- Define polled events by running the `pubmgr` tool.
- Define publish events for the Oracle Application Server InterConnect application in iStudio.
- Run the Oracle Application Server InterConnect application. This loads the Publishing Agent.
- If the events are configured to be manually triggered, you must launch the `pubmgr` and trigger the event. However, if the events are not manually triggered, the events will be triggered as soon as you start the application.

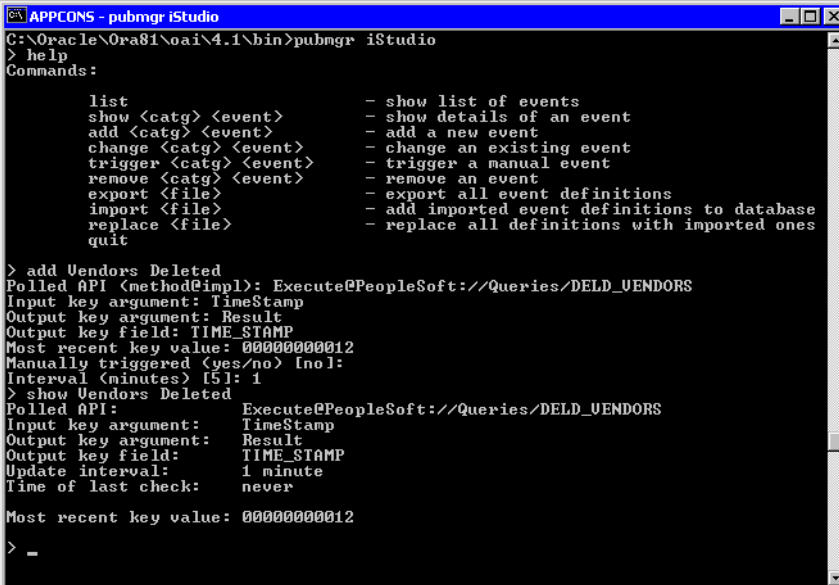
The Publishing Agent polls the system for the changes in data and reports the changes back to the Oracle Application Server InterConnect application.

Defining Polled Events using Publishing Manager

An event must be defined in the Publishing Manager to indicate its purpose. Use the following steps to define an event:

1. Change to the Installation directory\bin directory.
2. Type `pubmgr iStudio` and press **Enter**.

Figure 1–1 Creating an Event Definition using **PUBMGR**



```
APPCONS - pubmgr iStudio
G:\Oracle\Ora81\oai\4.1\bin>pubmgr iStudio
> help
Commands:

    list                - show list of events
    show <catg> <event> - show details of an event
    add <catg> <event>  - add a new event
    change <catg> <event> - change an existing event
    trigger <catg> <event> - trigger a manual event
    remove <catg> <event> - remove an event
    export <file>       - export all event definitions
    import <file>       - add imported event definitions to database
    replace <file>     - replace all definitions with imported ones
    quit

> add Vendors Deleted
Polled API <method@impl>: Execute@PeopleSoft://Queries/DELD_VENDORS
Input key argument: TimeStamp
Output key argument: Result
Output key field: TIME_STAMP
Most recent key value: 00000000012
Manually triggered <yes/no> [no]:
Interval <minutes> [5]: 1
> show Vendors Deleted
Polled API:           Execute@PeopleSoft://Queries/DELD_VENDORS
Input key argument:   TimeStamp
Output key argument:  Result
Output key field:    TIME_STAMP
Update interval:     1 minute
Time of last check:  never
Most recent key value: 00000000012
> -
```


3. Type `add` to add a new event definition. For example:

```
add Vendors Deleted
```

This creates a Vendors category if it does not already exist and adds the Deleted event definition to that category.

Note: Type `help` to get the list of commands and their syntax.

4. Enter the polled API. This is the business API that is polled periodically by the Publishing Engine. For example:

```
Execute@PeopleSoft://Queries/DELD_VENDORS
```

To find which URL to use, please refer to the iStudio section of your Adapter document. Follow the step up to the point where the Define Application View is populated. In the field for **Object Name** at the top of the screen, the URL will be displayed.

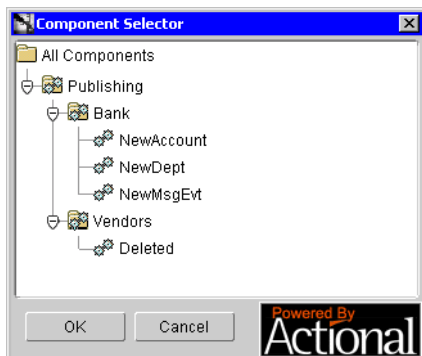
5. Enter the remaining parameters at the correct prompts.

See Also: ["Publishing Manager Commands"](#) on page 1-12

Defining a Published Event in iStudio

After you have defined the polled events for Publishing Engine, create the corresponding published events using iStudio.

Figure 1–2 *Browsing Polled Events in iStudio*



See Also: *Oracle Application Server InterConnect User's Guide*

Runtime Polling and Event Publishing using Publishing Engine

The publishing agent is responsible for the runtime polling and event publishing. It is started as part of the Oracle Application Server InterConnect application startup. Use the following steps to start this application:

1. Run the Oracle Application Server InterConnect application to load the Publishing Agent.
2. If the events are configured to be manually triggered, you must launch the `pubmgr` and trigger the event. However, if the events are not manually triggered, the events will be triggered as soon as you start the application.

The Publishing Agent polls the system for the changes in data and reports the changes back to the Oracle Application Server InterConnect application.

Use the following steps to use the Publishing Engine on a Unix machine:

1. Export the event definition by using the command `export filename` in the `pubmgr` on Windows.
2. Import the file by typing `import filename` in the `pubmgr`.

All commands for running `pubmgr` and defining events is the same on Unix as it is on Windows.

Publishing Manager Commands

Use the Publishing Manager to configure and monitor the Publishing Engine. Use the Publishing Manager to publish an event or message to monitor. The information set in the Publishing Manager is stored in the `config\pub.db` file created on activation of the Publishing Manager.

From and command line:

```
pubmgr [{-p | -r} <script>] [<profile>]
```

where:

- `-r`—Records a script containing each of the commands performed interactively during the execution of this session of the publishing manager.
- `-p`—Plays back a previously recorded script.

Each event, defined with the Publishing Manager, is given a category for organizational purposes. Examples of possible categories are Customers, Vendors, or PurchaseOrders. When using the Publishing Manager, the category must always be specified along with the event name.

[Table 1-2](#) describes each of the commands available within the Publishing Manager.

Table 1-2 Description of available commands

| Command | Syntax | Example | Description |
|---------|-------------------------------|--|---|
| List | list | Category Event Interval Vendor Created 1 minute Customers All Manual | Lists all defined events grouped by category. For each event the polling interval is also given. |
| Show | show <category> <event> | show Vendors Created Polled API: Execute@PeopleSoft://Queries/VE NDORS Input key argument: Vendor_ID Output key argument: Result Output key field: VENDOR_ID Update interval: 1 minute Time of last check: Fri Sep 01 15:19:28 2000 Most recent key value: 0000000044 | Shows the details of a particular event. For more information on the arguments in this example refer to the description of the Add command. |

Table 1–2 Description of available commands (Cont.)

| Command | Syntax | Example | Description |
|---------|---------------------------|---|---|
| Add | add <category> <event> | add Vendors Deleted Polled API (method@impl): Execute@PeopleSoft://Queries/DE LD_VENDORS Input key argument: TimeStamp Output key argument: Result Output key field: TIME_STAMP Most recent key value: 0000000012 Manually triggered (yes/no) [no]: Interval (minutes) [5]: | <p>Allows you to add a new event definition. The polled API is the business API that is polled periodically by the Publishing Engine.</p> <p>The input key argument, for internal key management, is the name of the argument that has a fixed value provided. This can be blank (input with a single period, ".") if there is no input value. For external key management this is the name of the argument used to pass in the stored key value.</p> <p>For external key management, if the output key is a standalone argument value, then the output key argument is the name of this argument and the output key field should be left blank (input with a single period, "."). If the output key is determined by examining a table, the output key argument is the name of the table argument and the output key field is the name of the key field within the table.</p> <p>The most recent key value, for internal key management, actually represents the fixed key value that is passed in to the input key argument each time the API is polled. For external key management this is the key value of the most recently synchronized data (data will be retrieved with key values after this entry).</p> <p>Polling of the business API can be either manually triggered (through the <code>Trigger</code> command) or automatically performed at a set interval (in minutes). The minimum polling interval is 1 minute. When entering values, press Enter to select the default value (shown between square brackets). However, unless an event has been set up to be triggered manually, it will be triggered at the interval entered in this field.</p> |

Table 1–2 Description of available commands (Cont.)

| Command | Syntax | Example | Description |
|---------|----------------------------------|--|--|
| Change | change <category> <event> | change Vendors Created Implementation:[Execute@ PeopleSoft://Queries/VENDORS]: Input key argument [Vendor_ID]: Output key argument [Result]: Output key field [VENDOR_ID]: Most recent key value [0000000044]: 0000000020 Manually triggered (yes/no) [no]: Interval (minutes) [1]: | Allows you to change existing event definitions. This can be used to update any of the fields (including the most recent key value if publishing of information needs to be repeated for any reason). The syntax of this command is identical to Add except that default values are provided for each field. Press Enter to select the default value. In order to enter a blank value instead of the default input a single period (".") followed by Enter . |
| Trigger | trigger <category> <event> | trigger Customers All | Allows you to trigger an event for polling. After trigger is used, the event is scheduled for polling within the next minute. You can use trigger to manually poll event (otherwise they will never be polled). Alternately, you can use trigger on interval polled events so that the polling occurs faster than it might otherwise. This is only useful on interval polled events if the polling interval is greater than 1 minute. |
| Remove | remove <category> <event> | remove Customers All | Removes the specified event definition and stored key information. There is no undo for the remove operation, so it is recommended to either export or show the definition before removing. |
| Export | export <filename> | export c:\Temp\events.xml | All values in the export format directly correspond to the values entered through Add. |
| Import | import <filename> | import c:\Temp\events.xml | Imports previously exported events definitions. If the events in the file do not already exist, they are created. If any of the events specified in the import file already exists, it is overwritten with the information found in the import file. This is typically used to load deployed event definitions. |

Table 1–2 Description of available commands (Cont.)

| Command | Syntax | Example | Description |
|---------|-----------------------|----------------------------------|--|
| Replace | replace <filename> | replace c:\Temp\events.xml | Similar to <code>Import</code> , it removes all existing event definitions and then imports the event definitions found in the file. <code>Replace</code> restores the Publishing Engine database to a previous state. |
| Help | help | help | Provides specific help using the following list of the possible commands. |
| quit | quit | quit C:\Oracle\ora81\oai\bin> | Quits the publishing manager. All changes are applied persistently as commands occur. <code>Quit</code> exits the manager. It does not affect the persistent store of the Publishing Engine. The Publishing Manager can be left running with commands applied periodically. However, if recording a script for later playback, the script is only written after a <code>Quit</code> command. |

Index

E

event publishing, 11
external key management, 6

I

internal key management, 5
istudio
 defining published events, 10

K

key management, 4
 external, 6
 internal, 5

P

publishing engine
 key management, 4
 supported platforms, 3
 using with OracleAS InterConnect, 7
 what is, 2
publishing manager
 defining events, 8
publishing manger
 commands, 12

R

runtime polling, 11

S

supported platforms, 3

