

Oracle Procedural Gateway® for APPC

Messages Guide

10g Release 2 (10.2) for UNIX and Microsoft Windows

B16213-01

June 2005

Oracle Procedural Gateway for APPC Messages Guide, 10g Release 2 (10.2) for UNIX and Microsoft Windows

B16213-01

Copyright © 2005, Oracle. All rights reserved.

Primary Author: Maitreyee Chaliha

Contributing Author: Shahrokh Talmoud

The Programs (which include both the software and documentation) contain proprietary information; 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. This document is not warranted to be 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.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth 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 we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Send Us Your Comments	v
Preface	vii
Intended Audience.....	vii
Documentation Accessibility	vii
Related Publications.....	viii
Conventions	viii
Storage Measurements	ix
Accessing Installed Documentation	ix
Oracle Services and Support.....	ix
1 TIP Exceptions	
Messages ORA-20700 to ORA-20704	1-1
2 Procedural Gateway Server Messages	
Messages PGA-20900 to PGA-20928	2-1
Messages PGA-20930 to PGA-21451	2-7
Message PGA-22001 to 22012 (for Gateway using TCP/IP).....	2-11
3 PGAU Messages	
Messages PGU-00001 to PGU-30028	3-1
Messages PGU-30030 to PGU-30637	3-10
Messages PGU-35002 to PGU-39999	3-18
Messages PGU-41000 to PGU-41119.....	3-26
Messages PGU-41120 to PGU-42042	3-38
4 pg4tcpmap Tool Messages	
Messages PGU-50001 to PGU-50101	4-1
Index	

Send Us Your Comments

Oracle Procedural Gateway for APPC Messages Guide, 10g Release 2 (10.2) for UNIX and Microsoft Windows

B16213-01

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. 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 about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us at the following e-mail address:

- Electronic mail: infodev_us@oracle.com
- FAX: (650) 506-7227. Attn: Server Technologies Documentation Manager
- Postal service:

Oracle Corporation
Server Technologies Documentation Manager
500 Oracle Parkway, Mailstop 4op11
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and e-mail address (optional).

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

Preface

This guide contains messages for all Oracle Procedural Gateway for APPC products. Use it with the *Oracle Database Server Messages* manual.

You can use this guide when you are installing or administering the gateway. You must understand the fundamentals of the operating system for your platform and procedural gateways before using this guide when installing or administering the gateway.

Intended Audience

This guide is intended for anyone responsible for installing, configuring and administering the gateway, and also for developers writing applications that access remote host databases through APPC.

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. 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.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Publications

There are two parts to the documentation set: the documentation specific to the Oracle Procedural Gateway for APPC and the general gateway documentation. Both documents come with the Oracle products you purchase. Use the general gateway documentation to learn about gateway concepts and the Oracle Procedural Gateway for APPC documentation to learn how to install, administer and use the gateway.

The documents included as a part of your product shipment are:

- *Oracle Procedural Gateway for APPC Messages Guide* 10g Release 2 (10.2)
- *Oracle Procedural Gateway for APPC User's Guide* for UNIX and Microsoft Windows

In addition, depending on your operating system, either of the following documents are included:

- *Oracle Procedural Gateway for APPC Installation and Configuration Guide for UNIX.*
- *Oracle Procedural Gateway for APPC Installation and Configuration Guide for Microsoft Windows*

Refer to the *Oracle Technical Publications Catalog and Price Guide* for a complete list of documentation provided for Oracle products.

Conventions

Examples of input and output for the gateway and Oracle environment are shown in a special font:

```
$ mkdir /ORACLE/your_name
```

All output is shown as it appears. For input, the list of conventions and their meanings are as follows:

- `example text`: Words or phrases, such as `mkdir` and `ORACLE`, must be entered exactly as spelled and in the letter case shown. In this example, `mkdir` must be entered in lowercase letters and `ORACLE` in uppercase letters.
- *italics monospace text*: Italicized uppercase or lowercase, such as *your_name*, indicates that you must substitute a word or phrase, such as the actual directory name.
- **BOLD text** or **bold italic TEXT**: Bold words or phrases refer to a file or directory structure, such as a directory, path, or file ID.
- ... : Ellipses indicate that the preceding item can be repeated. You can enter an arbitrary number of similar items.
- { }: Curly braces indicate that one of the enclosed arguments is required. Do not enter the braces themselves.
- | : Vertical lines separate choices.

- []: Square brackets enclose optional clauses from which you may choose one or none. Do not enter the brackets.

Other punctuation, such as commas, quotation marks, or the pipe symbol (|), must be entered as shown unless otherwise specified. Directory names, file IDs, and so on appear in the required letter case in examples. The same convention is used when these names appear in text, and the names are highlighted in **bold**. The use of *italics* indicates that those portions of a file ID that appear in *italics* can vary.

Gateway commands, file IDs, reserved words, and keywords appear in uppercase in examples and text. UNIX commands, environment variables, and keywords appear in the required letter case in examples and text. Reserved words and keywords must always be entered as shown, because they have reserved meanings within the Oracle system.

Storage Measurements

Storage measurements use the following abbreviations:

- KB, for kilobyte, which equals 1,024 bytes
- MB, for megabyte, which equals 1,048,576 bytes
- GB, for gigabyte, which equals 1,073,741,824 bytes

Accessing Installed Documentation

Following documents are included with the product.

Documentation for Oracle Procedural Gateway for APPC for UNIX and Windows operating system

Documentation for this product includes this guide and the *Oracle Procedural Gateway for APPC, 10g Release 2 (10.2) for UNIX and Microsoft Windows*, and the *Oracle Procedural Gateway for APPC Installation and Configuration Guide, 10g Release 2 (10.2) for UNIX and Windows*.

To access the documentation in HTML and PDF formats, use a browser to open the top level of the Gateway Documentation Software included on your installation media. This level contains links to the product and your Microsoft Windows or UNIX-specific documentation.

Oracle Product Documentation

Oracle Database product documentation is on the Oracle Database Generic Documentation installation media. Instructions for accessing and installing the documents on the installation media are in the README file in the top level directory of the installation media.

Oracle Services and Support

The Oracle corporate Web page is at the following address:

<http://www.oracle.com>

Oracle offers a wide range of services to help facilitate corporate system solutions, including Oracle Education courses, Oracle Consulting services, and Oracle Support

Services, from the Web site. In addition, Oracle provides free trial software, updates on Oracle products and services, and technical brochures and data sheets.

Oracle Support Services

Technical Support registration and contact information worldwide is available at the following address:

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

At the Oracle support site, you will find templates to help you prepare information about your problem before you call so that you may be helped more quickly. You will also need your CSI number (if applicable) or complete contact details, including any special project information.

Oracle Technology Network

Oracle Technology Network (OTN) delivers all product documentation, as well as technical papers, code samples, self-service developer support, and the Oracle key developer products to enable rapid development and deployment of applications built on Oracle technology.

Register with the OTN at:

<http://www.oracle.com/technology/membership/index.html>

All Oracle product documentation can be found at:

<http://www.oracle.com/technology/documentation/index.html>

OracleMetaLink

OracleMetaLink is the Oracle Web service for technical information. Members of *OracleMetaLink* can search for updates, alerts, patches, and other information about products, releases, and operating systems, or set preferences to be notified automatically of new information. *OracleMetaLink* offers a variety of services to assist in setting up and administrating Oracle products, including procedures, scripts, commentary, and tuning and configuration best-practices bulletins.

Log on to *OracleMetaLink* before installing or administrating your product to search for up to date information about Oracle Database 10g Release 2 (10.2) for UNIX or Microsoft Windows.

In addition, *OracleMetaLink* offers forums for information sharing among Oracle customers and direct communication with Oracle Support Services. Use your Support Access Code (SAC) number to register. *OracleMetaLink* is available to Product Support Customers at no extra cost.

Sign up for free membership for this service at the following site:

<http://www.oracle.com/technology/membership/index.html>

Oracle Products and Other Documentation

The Oracle Store has links to U.S. and other countries at the following Web site:

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

Customer Service

Oracle Support Services contacts are listed at the following Web site:

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

Support for Hearing and Speech Impaired Customers

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week.

- For technical questions, call:
1.800.446.2398
- For non-technical questions, call:
1.800.464.2330

Education and Training

Training information and worldwide schedules are available from:

<http://education.oracle.com>

TIP Exceptions

This chapter lists the TIP exceptions issued by the PGAU-generated TIPs and provides a possible cause and the recommended action for you to take for each message.

This chapter contains the following section:

- [Messages ORA-20700 to ORA-20704](#) on page 1-1

Messages ORA-20700 to ORA-20704

ORA-20700 PGA_TIP: repeating group limits exceeded

Cause: The repetition count in a repeating group is outside its limits as defined in the PG DD for the group. The actual data being converted and exchanged through the TIP contains a count field with an invalid number.

Action: Generate the TIP with data conversion and data exchange tracing as required or rerun the application with tracing enabled. Analyze the resulting TIP and gateway server traces. Supported customers can contact Oracle Support Services for assistance.

ORA-20701 PGA_TIP: sent data length invalid *cname pname*, expected *elen*, converted *clen*

Cause: TIP data lengths for the, *cname pname*, send parameter did not correspond. One of the indicated lengths, *elen* or *clen*, is incorrect.

Action: Generate the TIP with data conversion and data exchange tracing as required or rerun the application with tracing enabled. Analyze the resulting TIP and gateway server traces. Supported customers can contact Oracle Support Services for assistance.

ORA-20702 PGA_TIP: received data length invalid *cname pname*, expected *elen*, received *rlen*, converted *clen*

Cause: TIP data lengths for the received parameter, *cname pname*, did not correspond. One of the indicated lengths, *elen*, *rlen*, or *clen*, is incorrect.

Action: Generate the TIP with data conversion and data exchange tracing as required or rerun the application with tracing enabled. Analyze the resulting TIP and gateway server traces. Supported customers can contact Oracle Support Services for assistance.

ORA-20703 PGA_TIP: pipe send error, *rc* = *rc*

Cause: The indicated error, *rc*, occurred on a DBMS_PIPE send call. The call was made because TIP tracing was enabled. If the value of *rc* equals to 1, then the 60-second pipe wait time elapsed. This is usually because the TIP trace pipe has overflowed at the inlet due to the trace message inflow exceeding 16K.

Action: Ensure that the `rtrace.sql` procedure is run often enough to keep the TIP trace pipe from filling and that the server output size is sufficient to hold the trace stream between `rtrace` calls.

See Also: For more information, refer to Chapter 6, "Problem Determination," in *Oracle Procedural Gateway for APPC User's Guide*. Supported customers can contact Oracle Support Services for assistance.

ORA-20704 PGA_TIP: tranuse value cannot be shared

Cause: The `tranuse` value passed to a TIP function was of the type that is used for unshared TIP conversations that are associated with TIPS generated prior to release 3.4.0. This happens when a pre-3.4.0 version of a TIP is used to initiate a conversation, and its `tranuse` value is subsequently passed to a post-3.4.0 TIP to use both TIPS in a shared conversation.

Action: Regenerate the pre-3.4.0 TIP to cause it to pick up a new function that supports the TIP conversation sharing feature. Do not initiate the conversation with (or otherwise use) the old TIP with the new TIP. Do not call the new TIP for the active conversation. Pre-3.4.0 TIPS cannot be used with post-3.4.0 TIPS for shared conversations.

Procedural Gateway Server Messages

This chapter lists the messages issued by the Oracle Procedural Gateway for APPC server and provides a possible cause and recommended action for each message. In addition to these messages, there are gateway messages, prefixed with GTW, that are documented in *Oracle Open Gateways Guide for SQL-Based and Procedural Gateways*.

This chapter contains the following sections:

- [Messages PGA-20900 to PGA-20928](#) on page 2-1
- [Messages PGA-20930 to PGA-21451](#) on page 2-7
- [Message PGA-22001 to 22012 \(for Gateway using TCP/IP\)](#) on page 2-11

Messages PGA-20900 to PGA-20928

PGA-20900 unable to obtain *n* bytes of storage for *description*

Cause: A memory shortage occurred.

Action: Ensure that your system has enough available memory to support the number of concurrent users you are running.

PGA-20901 internal gateway error: [*arg1*] [*arg2*] [*arg3*] [*arg4*] [*arg5*]

Cause: An Internal error occurred.

Action: Regenerate the error with debugging enabled and generate a log file. Supported customers should contact Oracle Support Services for assistance.

PGA-20905 invalid conversation id: no active conversations were found

Cause: No APPC conversations were active for the user.

Action: Check if the application does not calling the PL/SQL TIP routines out of sequence.

PGA-20906 invalid conversation id: no matching conversation was found

Cause: Conversation ID received from the caller was not a valid active conversation ID.

Action: Check if the application is not calling the PL/SQL TIP routines out of sequence.

PGA-20907 preceding error occurred during gateway func processing

Cause: An error occurred while processing the Oracle specified function, *func*. This message is preceded by additional messages that provide more information about the error.

Action: Refer to the preceding messages that provide to determine the course of action.

Locate Error Message PGA-20910 for Your Platform From This List

PGA-20910 communication error: CPI-C *func* failed, *rc* = *rc*, *errno* = *errno*

Cause: (on AIX systems) An unexpected communication error occurred while executing the specified CPI-C function (*func*). The CPI-C function return code (*rc*) and system error number (*errno*) are provided in the message text. CPI-C function return codes and system error numbers are described in the vendor documentation. Also, the CPI-C return codes are available in the `/usr/include/cmc.h` header file, and the AIX SNA system error numbers can be found in the `/usr/include/luxsna.h` header file.

[Table 2-1](#) lists some commonly received *rc* and *errno* combinations on AIX and their possible causes.

Action: Determine the cause of the communications error, correct it, and rerun the transaction.

Table 2-1 Error Message PGA-20910 for AIX Systems

rc	errno	Possible Causes
1	22	The partner LU location profile for the target LU is either missing or incorrect. It is also possible that or the target LU is not active or does not have SNA communications enabled.
1	25	The LU name is not defined on the target system. This name is the fully qualified LU name specified in either the Side Information Profile or the Partner LU Location Profile.
6	133	A security violation occurred on the target system. Either the user ID or password is not valid on that system, or the user ID is not authorized to execute the requested transaction.
9	127	The transaction program requested is not defined to the target LU.
10	147	The transaction program requested is defined to the target LU, but could not be found.
17	121	The transaction program terminated abnormally on the target LU, or issued a DEALLOCATE_ABEND.
19	9	Either the target LU name or alias is not defined locally to SNA Server, the mode name is not defined locally to SNA Server, or the mode name is not defined to the target LU.
20	125	The SNA session on which the conversation was running has been terminated, or which SNA communication package is shutting down.
20	160	SNA Server is not running.

PGA-20910 communication error: CPI-C *func* failed, *rc* = *rc*, *errno* = *errno*

Cause: (on HP-UX) An unexpected communication error occurred while executing the specified CPI-C function (*func*). The CPI-C function return code (*rc*) and system error number (*errno*) are provided in the message text. The CPI-C function return codes and system error numbers are described in the vendor documentation.

Commonly received return codes and their possible causes on hp-ux are listed in [Table 2-2](#).

Action: Determine the cause of the communications error, correct it, and rerun the transaction.

Table 2–2 Error Message PGA-20910 for HP-UX

rc	errno	Possible Cause(s)
2	N/A	The target LU is not active or has not established communications with the SNA communications package.
6	N/A	A security violation occurred on the target system. Either the user ID or password is not valid on that system, or the user ID is not authorized to execute the requested transaction on the target OLTP.
9	N/A	The transaction program requested is not defined to the target OLTP.
10	N/A	The transaction program requested is defined to the target OLTP, but the OLTP could not be found.
17	N/A	The transaction program terminated abnormally on the target OLTP or issued a DEALLOCATE_ABEND.
19	N/A	The mode name specified was not defined in the mode table for the target LU, or the LU name specified does not exist, or the LU name specified is not defined to the SNA communication packages.
26	N/A	The SNA communication package is not active, or the local node or link station is not active.

PGA-20910 communication error: CPI-C *func* failed, rc = rc, errno = errno

Cause: (on Solaris) An unexpected communication error occurred while executing the specified CPI-C function (*func*). The CPI-C function return code (*rc*) and system error number (*errno*) are provided in the message text. For the SNA communication package, CPI-C function return codes are described in the Solaris vendor documentation.

Table 2–3 lists commonly received *rc* and *errno* combinations, and their possible causes on Solaris.

Action: Determine the cause of the communications error, correct it, and rerun the transaction.

Table 2–3 Error Message PGA-20910 for Solaris

rc	errno	Possible Cause(s)
6	5	A security violation occurred on the target system. Either the user ID or password is not valid on that system, or the user ID is not authorized to execute the requested transaction on the target OLTP.
9	0	The transaction program requested is not defined to the target OLTP.
17	0	The transaction program terminated abnormally on the target OLTP, or issued a DEALLOCATE_ABEND.

PGA-20910 communication error: CPI-C *func* failed, rc = rc, errno = errno

Cause: (on Microsoft Windows) An unexpected communication error occurred while executing the specified CPI-C function (*func*). The CPI-C function return code (*rc*) and system error number (*errno*) are provided in the message's text. The CPI-C function return codes are described in the vendor documentation. Also, the CPI-C return codes are contained in the C:\bkoffice\include\wincpic.h header file. The system error number is

meaningless for the Host Integration Server on Microsoft Windows and is usually 0.

Table 2–4 lists some commonly received `rc` or `errno` messages, and their possible causes.

Action: Determine the cause of the communication error, correct it, and rerun the transaction.

Table 2–4 Error Message PGA-20910 for Microsoft Windows

<code>rc</code>	<code>errno</code>	Possible Cause(s)
2	0	Either the target LU name is not defined on the target system, or the mode name is not defined on the target system.
6	0	A security violation occurred on the target system. Either the user ID or password is not valid on that system, or the user ID is not authorized to execute the requested transaction.
9	0	The transaction program requested is not defined to the target LU.
10	0	The transaction program requested is defined to the target LU but could not be found.
17	0	The transaction program terminated abnormally on the target LU or issued a <code>DEALLOCATE _ABEND</code> .
19	0	The target LU is not active or has not established communication with the Microsoft Host Integration Server.
20	0	Either the target LU name or alias is not defined locally to the Microsoft Host Integration Server, or the mode name is not defined locally to the Microsoft Host Integration Server.

If you use the IBM Communication Server, then the error message numbers and return codes that you get are different from those listed. Refer to the `wincpic.h` file for more information about error message numbers.

PGA-20911 update transaction is already active with TP *tpname* at LU *luname*

Cause: Transaction `tpname` has already been started at LU `luname` by a `PGAINIT` call with `synclevel` set to 1 or 2. Only one transaction at `synclevel` 1 or 2 is allowed at any given time.

Action: The new transaction does not start. Change your application to start the second update transaction after the first one has completed.

PGA-20912 send and receive buffer lengths cannot both be zero

Cause: Both the send and receive buffer lengths passed to `PGAXFER` were zero. This is invalid.

Action: Check that the `PGDL` and `COBOL` record descriptions that are used to define the transaction to `PGAU` are synchronized with the transaction program and that the `PL/SQL` TIP was generated by `PGAU` using the correct definitions. If modifications have been made to the generated `PL/SQL` code, then check that they are correct.

PGA-20914 send buffer length of *len* exceeds actual send buffer size of *size*

Cause: The send buffer length (`len`) passed to `PGAXFER` was larger than the actual size (`size`) of the send buffer passed to `PGAXFER`.

Action: Check that the `PGDL` and `COBOL` record descriptions that are used to define the transaction to `PGAU` are synchronized with the transaction program and that the `PL/SQL` TIP was generated by `PGAU` by using the correct definitions. If

modifications have been made to the generated PL/SQL code, then check that they are correct.

PGA-20915 receive buffer length of *len* exceeds the maximum of *max*

Cause: The receive buffer length (*len*) passed to PGAXFER was larger than the maximum allowed length (*max*).

Action: Check that the PGDL and COBOL record descriptions that are used to define the transaction to PGAU are synchronized with the transaction program and that the PL/SQL TIP is generated by PGAU by using the correct definitions. If modifications have been made to the generated PL/SQL code, then check that they are correct.

PGA-20916 send count is *count* but only *num* send lengths were specified

Cause: The first value (*count*) in the send lengths array passed to PGAXFER specified more elements than the send lengths array contained. Only (*num*) elements were found in the array.

Action: Check that the PGDL and COBOL record descriptions that are used to define the transaction to PGAU are synchronized with the transaction program and that the PL/SQL TIP is generated by PGAU by using the correct definitions. If you have modified the generated PL/SQL code, then check that the modifications are correct.

PGA-20917 receive count is *count* but only *num* receive lengths were specified

Cause: The first value (*count*) in the receive lengths array passed to PGAXFER specified more elements than the receive lengths array contained. Only (*num*) elements were found in the array.

Action: Check that the PGDL and COBOL record descriptions that are used to define the transaction to PGAU are synchronized with the transaction program and that the PL/SQL TIP is generated by PGAU by using the correct definitions. If you have modified the generated PL/SQL code, then check that the modifications are correct.

PGA-20918 send length number *num* is *len* but only *bytes* bytes are left in the buffer

Cause: The length (*len*) specified in send lengths array element number (*num*) exceeded the number of bytes (*bytes*) of data remaining in the send buffer.

Action: Check that the PGDL and COBOL record descriptions that are used to define the transaction to PGAU are synchronized with the transaction program and that the PL/SQL TIP is generated by PGAU by using the correct definitions. If modifications have been made to the generated PL/SQL code, then check that they are correct.

PGA-20919 receive length number *num* is *len* but only *bytes* bytes are left in the buffer

Cause: The length (*len*) specified in receive lengths array element number (*num*) exceeded the number of bytes (*bytes*) of space remaining in the receive buffer.

Action: Check that the PGDL and COBOL record descriptions that are used to define the transaction to PGAU are synchronized with the transaction program and that the PL/SQL TIP is generated by PGAU by using the correct definitions. If modifications have been made to the generated PL/SQL code, then check that they are correct.

PGA-20920 unexpected request-to-send received

Cause: The remote transaction program requested to send data when the gateway was still sending data.

Action: Check that the PGDL and COBOL record descriptions that are used to define the transaction to PGAU are synchronized with the transaction program and that the PL/SQL TIP is generated by PGAU by using the correct definitions. If modifications have been made to the generated PL/SQL code, then check that they are correct.

PGA-20921 buffer overflow on receive: requested *num* bytes, received *len* bytes

Cause: The remote transaction program sent (*len*) bytes of data when the gateway was expecting to receive only (*num*) bytes.

Action: Check that the PGDL and COBOL record descriptions that are used to define the transaction to PGAU are synchronized with the transaction program and that the PL/SQL TIP is generated by PGAU by using the correct definitions. If modifications have been made to the generated PL/SQL code, then check that they are correct.

PGA-20922 no data available to receive: *num* bytes were requested

Cause: The remote transaction program either requested to receive or deallocate the conversation, but the gateway is still expected to receive more data.

Action: Check that the PGDL and COBOL record descriptions that are used to define the transaction to PGAU are synchronized with the transaction program and that the PL/SQL TIP is generated by PGAU by using the correct definitions. If modifications have been made to the generated PL/SQL code, then check that they are correct.

PGA-20923 unexpected status, *stat*, received after *func*

Cause: An unexpected status code (*stat*) was received from the remote transaction program following the CPI-C (*func*) call. The descriptive name of the status code is (*desc*). CPI-C *func* function receives the status.

Action: This is usually a problem with the remote transaction program or a network problem. The status codes for each CPI-C function call are documented in the platform-specific SNA software documentation for your system. documentation.

See Also: Refer to the System Requirements chapter of the *Oracle Procedural Gateway for APPC Installation and Configuration Guide* for your operating system for titles of SNA software

PGA-20924 unable to enter send state for normal deallocate, state = *state* (*desc*)

Cause: The gateway was unable to enter send state to perform a normal deallocation. The conversation state is (*state*), and its descriptive name is (*desc*).

Action: This is usually a problem with the remote transaction program or with the network. Check the remote system for diagnostic information. The gateway performs an abnormal deallocation after this error has been encountered.

PGA-20925 missing LU/TP/Mode name parameter and no side info profile specified

Cause: PGAINIT was called with no Side Information profile name specified, but the LU name, TP name, and Mode name parameters were not all filled in with non blank values.

Action: Check the PGDL used to define the transaction program to PGAU. Ensure either a valid Side Information profile name is specified (SIDEPROFILE keyword), or that the LU name, TP name, and Mode name (LUNAME, TPNAME, LOGMODE keywords) are all specified with valid values. If no Side Information

profile is specified, then the LU name, TP name, and Mode name parameters are all required.

PGA-20926 userid *uid* length of *len* is invalid; maximum length is *max*

Cause: The user ID (*uid*) was too long (*len*) to be used with APPC conversation security. The maximum allowable length for the user ID is *max* characters. This can occur only when the gateway initialization parameter `PGA_SECURITY_TYPE` is set to either `SAME` or `PROGRAM`.

Action: Change the user ID to a shorter value on both the client system and the system where the remote transaction program is run. This restriction is imposed by SNA and APPC, and not by the gateway.

PGA-20927 password length of *len* is invalid; maximum length is *max*

Cause: The password length (*len*) was too long to be used with APPC conversation security. The maximum allowable length for the password is *max* characters. This can occur only when the `PGA_SECURITY_TYPE` gateway initialization parameter is set to `PROGRAM`.

Action: Change the password to a shorter value on both the client system and the system where the remote transaction program is run. This restriction is imposed by SNA and APPC, and not by the gateway.

PGA-20928 cannot open parameter file %s

Cause: The gateway could not locate or open the specified file.

Action: Ensure that the file exists in a location expected by the gateway and can be opened.

Messages PGA-20930 to PGA-21451

PGA-20930 invalid SYNCLEVEL, *sync*, specified; valid range is *min:max*

Cause: The `synclevel`, *sync*, passed to `PGAINIT` was not a valid value. This value is specified by the `SYNCLEVEL` keyword in the `DEFINE TRANSACTION` statement that is used to define the transaction to `PGAU`.

Action: The value for `SYNCLEVEL` must fall within the range, *min:max*. Correct the value specified in the `DEFINE TRANSACTION` statement for the `SYNCLEVEL` keyword and regenerate the PL/SQL TIP using `PGAU`.

PGA-20931 send buffer length of *len* exceeds the maximum of *max*

Cause: The send buffer length, (*len*), passed to `PGAXFER` was larger than the maximum allowed, *max*.

Action: Check that the `PGDL` and `COBOL` record descriptions used to define the transaction to `PGAU` do not define any data items larger than the maximum size allowed by APPC. Correct the data items in error and regenerate the PL/SQL TIP using `PGAU`.

PGA-20932 invalid function code, *func*, passed to `pgatctl`

Cause: The function code, *func*, passed to the `PGATCTL` function was invalid.

Action: Correct the function code in the PL/SQL procedure, recompile it, and retry the operation.

PGA-20933 invalid value specified by initialization parameter *keyword=value*

Cause: The keyword parameter, *keyword=value*, specifies an invalid value.

Action: Refer to the *Oracle Procedural Gateway for APPC Installation and Configuration Guide* documenting gateway initialization parameters for the keyword and correct the error.

PGA-20934 side information profile *profile* not defined; cannot establish conversation

Cause: The Side Information profile (*profile*) was not defined to the SNA software. Either the profile name was misspelled, or no profile has been defined.

Action: Correct the profile name if it was misspelled, or define the profile defined to the SNA software if it was not already defined.

PGA-20935 sync level *sync* is not allowed when `PGA_CAPABILITY=cap`

Cause: The sync level parameter that was (*sync*) passed to PGAINIT was incompatible with the setting of the `PGA_CAPABILITY` gateway initialization parameter (*cap*).

Action: If the sync level passed to PGAINIT is correct, then the `PGA_CAPABILITY` gateway initialization parameter must be changed to allow the desired sync level to be supported. If the sync level is not correct, then the TIP should be changed to specify the correct sync level in the call to PGAINIT.

If the TIP was generated by PGAU, then the `SYNLEVEL` keyword of the `DEFINE TRANSACTION` statement should be changed to specify the correct sync level and the TIP should be regenerated. In the following list, each `PGA_CAPABILITY` value setting is followed by the number of the sync levels that are valid for that setting:

- `READ_ONLY` or `RO`: 0 sync levels allowed
- `SINGLE_SITE` or `SS`: 0, 1 sync levels allowed
- `COMMIT_CONFIRM` or `CC`: 0,1 sync levels allowed

PGA-20936 send buffer length is *len* but no send lengths were specified

Cause: The send buffer length passed to PGAXFER was (*len*), but the send lengths array was either null or contained a send count of zero.

Action: Check that the PGDL or COBOL record descriptions that are used to define the transaction to PGAU are synchronized with the transaction program and that the PL/SQL TIP is generated by PGAU by using the correct definitions.

PGA-20937 receive buffer length *len* but no receive lengths were specified

Cause: The receive buffer length passed to PGAXFER was (*len*), but the receive lengths array was either null or contained a receive count of zero.

Action: Check that the PGDL or COBOL record descriptions that are used to define the transaction to PGAU are synchronized with the transaction program and that the PL/SQL TIP is generated by PGAU by using the correct definitions.

PGA-20938 send lengths array too small (*len*) to contain a valid send count

Cause: The send lengths array passed to PGAXFER was too small to contain a valid send item count. The actual length of the send lengths array was (*len*). The minimum length of the send lengths array is 4 bytes.

Action: If the TIP was generated by PGAU, then ensure that it was modified correctly. If the TIP was not modified, then supported customers should contact Oracle Support Services for assistance. If the TIP was modified or was not generated by PGAU, then correct the send lengths array passed to PGAXFER.

PGA-20939 receive lengths array too small (*len*) to contain a valid receive count

Cause: The receive lengths array passed to PGAXFER was too small to contain a valid receive item count. The actual length of the receive lengths array was (len) . The minimum length of the receive lengths array is 4 bytes.

Action: If the TIP was generated by PGAU, then ensure that it was modified correctly. If the TIP was not modified, then supported customers should contact Oracle Support Services for assistance. If the TIP was modified or was not generated by PGAU, then correct the receive lengths array passed to PGAXFER.

PGA-20945 unable to log on to Oracle for transaction logging

Cause: The gateway server was unable to connect to the Oracle server where the transaction log table is stored. An Oracle server message follows this message.

Action: Ensure that the Oracle server and its TNS listener are both operational. Also, check the PGA_LOG_DB, PGA_LOG_USER, and PGA_LOG_PASS parameters in the gateway initialization file and ensure that they specify the correct database string, user ID, and password, respectively.

PGA-20947 unable to bind variable *var* for transaction logging

Cause: The gateway server was unable to bind a variable (*var*) for use in performing transaction logging functions. An Oracle server message follows this message.

Action: Ensure that the Oracle server and its TNS listener are both operational. Also, check the PGA_LOG_DB, PGA_LOG_USER, and PGA_LOG_PASS parameters in the gateway initialization file, and ensure that they specify the correct database string, user ID, and password, respectively.

PGA-20948 unable to prepare the statement for transaction logging

Cause: The gateway server was unable to prepare the statement to be used to call the transaction logging PL/SQL stored procedure. An Oracle server message follows this message.

Action: Ensure that the Oracle server and its TNS listener are both operational. Also, check the PGA_LOG_DB, PGA_LOG_USER, and PGA_LOG_PASS parameters in the gateway initialization file, and ensure that they specify the correct database string, user ID, and password, respectively. Verify that the transaction logging PL/SQL procedure is properly installed on the Oracle server under the user ID and password specified by PGA_LOG_USER and PGA_LOG_PASS and that it is executable.

PGA-20949 unable to define variable *var* for transaction recovery

Cause: The gateway server was unable to define the variable (*var*) to be used for accessing the transaction log table during recovery processing. An Oracle server message will follow this message.

Action: Ensure that the Oracle server and its TNS listener are both operational. Also check the PGA_LOG_DB, PGA_LOG_USER, and PGA_LOG_PASS parameters in the gateway initialization file, and ensure that they specify the correct database string, user ID, and password, respectively.

PGA-20950 unable to perform *oper* on transaction log

Cause: The gateway server was unable to insert, update, or delete a pending transaction row in the transaction log table. An Oracle server message follows this message.

Action: Ensure that the Oracle server and its TNS listener are both operational. Also check the PGA_LOG_DB, PGA_LOG_USER and PGA_LOG_PASS parameters in the gateway initialization file, and ensure that they specify the correct database

string, user ID, and password, respectively. Verify that the transaction logging PL/SQL stored procedure and the transaction log table are properly installed into the Oracle server under the user ID and password specified by PGA_LOG_USER and PGA_LOG_PASS.

PGA-20995 communication error: func failed, rc =rc, errno = errno

Cause: An unexpected communications error occurred while executing the AIX SNA Server API function (*func*). The function return code is *rc* and the system error number is *errno*. The AIX SNA Server LU6.2 API function return codes and system error numbers are described in the vendor documentation. The SNA system error numbers can also be found in the `/usr/include/luxsna.h` header file on AIX systems.

Table 2-5 lists commonly received *rc* and *errno* combinations and their possible causes.

Action: Determine the cause of the communications error, correct it, and rerun the transaction. If you need assistance, then contact your system administrator.

Table 2-5 Error Message PGA-20995 on AIX Systems

rc	errno	Possible Cause(s)
-1	112	When received at conversation startup time, either the target LU is not active or does not have SNA communications enabled, or the target LU name is not defined on the target system. The target LU name is the fully qualified LU name specified in either the side information profile or the Partner LU Location Profile. When received during an active conversation, the SNA session on which the conversation was running gets terminated.
-1	113	A conversation was requested at synclevel 2, but the required LU6.2 Resource Recovery Manager, PG4ARRM, is not active on the Local LU.
-1	121	The transaction program terminated abnormally on the target LU or issued a DEALLOCATE_ABEND.
-1	123	The mode name specified is either not locally defined to SNA Server, or it is not defined to the target LU.
-1	127	The transaction program requested is not defined to the target LU.
-1	133	A security violation occurred on the target system. Either the user ID or password is not valid on that system, or the user ID is not authorized to execute the requested transaction.
-1	146	The target LU name or alias is not locally defined to SNA Server.
-1	147	The transaction program requested is defined to the target LU but could not be found.
-1	160	The SNA communication package is not active.

PGA-20999 SIGDANGER received from system - all conversations deallocated

Cause: The system sent a SIGDANGER signal to the server indicating that there is a shortage of paging space. All currently active conversations are deallocated normally.

Action: Contact your system administrator.

See Also: Refer to the *Oracle Database Server Messages* for information about this Oracle server message

Message PGA-22001 to 22012 (for Gateway using TCP/IP)

PGA-22001 remote host name is *null*.

Cause: TCP/IP protocol has returned `null` for the remote host name.

Action: Verify the remote host name by using the `pg4tcpmap` tool and rerun your PL/SQL statement. If the problem cannot be resolved, supported customers should contact Oracle Support Services.

PGA-22002 communication error: TCP/IP *func* failed, *rc* = *rc*

Cause: The TCP/IP function (`func`) has failed with the return code of `rc`.

Action: If the problem cannot be resolved, then supported customers should contact Oracle Support Services.

PGA-22003 communication error: TCP/IP *func* failed, *rc* = *rc*, for *parm*

Cause: An unexpected communications error occurred while running the specified TCP/IP function (`func`). The TCP/IP function return code (`rc`) and the function (`func`) parameter is (`parm`).

Action: If the problem cannot be resolved, then supported customers should contact Oracle Support Services.

PGA-22004 length of the total message value *len* not within valid range (*min:max*)

Cause: The length (`len`) for the value is not within the valid range (`min:max`).

Action: If the problem cannot be resolved, then supported customers should contact Oracle Support Services.

PGA-22005 invalid socket file descriptor: no active conversations were found

Cause: There were no TCP/IP conversations active for the user.

Action: Check whether the application is calls the PL/SQL TTP routines out of sequences.

PGA-22006 parameter *parm* is not specified

Cause: A Parameter (`parm`) is missing from the gateway initialization file.

Action: Add the missing parameter to the gateway initialization file.

PGA-22007 user ID *uid* length of *len* is invalid; maximum length is *max*

Cause: The user ID (`uid`) is too long (`len`) to be used. The maximum length for the user ID is `max` characters.

Action: Change the user ID to a shorter value for OCI logon to the table `PGA_TCP_IMSC`.

PGA-22008 password length of *len* is invalid; maximum length is *max*

Cause: The password is too long (`len`) to be used. The maximum length for the password is `max`.

Action: Change the password to a shorter value for OCI logon to the `PGA_TCP_IMSC` table.

PGA-22009 database name *db name* length of *len* is invalid; maximum length is *max*

Cause: The database name (`db name`) is too long (`len`) to be used. The maximum allowable length of the database name is `max`.

Action: Change the database name to a shorter value for OCI logon to the `PGA_TCP_IMSC` table.

PGA-22010 userid *uid* length of *len* is invalid; maximum length is *max*

Cause: The user ID (*uid*) is too long (*len*) to be used with TCP/IP RACF conversation security. This can occur only when the `PGA_SECURITY_TYPE` gateway initialization parameter is set to either `SAME` or `PROGRAM`.

Action: Change the user ID to a shorter value on both the client system and the system where the remote transaction program is run. The maximum length for the user ID is *max* characters. This restriction is imposed by RACF, and not by the gateway.

PGA-22011 password length of *len* is invalid; maximum length is *max*

Cause: The password is too long (*len*) to be used with TCP/IP RACF conversation security. This can occur only when the `PGA_SECURITY_TYPE` gateway initialization parameter is set to `PROGRAM`.

Action: Change the password to a shorter value on both the client system and the system where the remote transaction program is run. The maximum length for the password is *max* characters. This restriction is imposed by RACF, and not by the gateway.

PGA-22012 IMS Connect error: return code = *rc*, reason code =*rsc*

Cause: An unexpected IMS Connect error occurred while executing the `recv` function. The return code (*rc*) and reason code (*rsc*) are provided in the message text.

For more information, refer to the `OLTP SDSF` log and the IMS Connect error message that starts with `HWS`. The messages are documented in the IBM IMS Connect Guide and Reference Manual

Action: Determine the cause of the error, correct it, and rerun the transaction.

PGA-22014 transaction code length of *len* exceeds the maximum of *max*

Cause: The transaction code length (*len*) passed to `PGAINIT` or `PGAINIT_SEC` was larger than the maximum allowed length (*max*).

Action: Check that the `PGDL` and `COBOL` record descriptions used to define the transaction to `PGAU` are synchronized with the transaction program and that the `PL/SQL TIP` was generated by `PGAU` using the correct definitions. If modifications have been made to the generated `PL/SQL` code, then ensure that they are correct.

See Also: For more information, refer to the *Oracle Procedural Gateway Installation and Configuration Guide* for the required gateway initialization parameters for gateways using TCP/IP

PGAU Messages

This chapter lists the messages issued by the Procedural Gateway Administration Utility (PGAU) and provides a possible cause and recommended action for each message.

This chapter contains the following sections:

- [Messages PGU-00001 to PGU-30028](#) on page 3-1
- [Messages PGU-30030 to PGU-30637](#) on page 3-10
- [Messages PGU-35002 to PGU-39999](#) on page 3-18
- [Messages PGU-41000 to PGU-41119](#) on page 3-26
- [Messages PGU-41120 to PGU-42042](#) on page 3-38

Messages PGU-00001 to PGU-30028

PGU-00001 through PGU-00096

Cause: These are informational messages.

Action: No action is required.

PGU-00100 invalid SPOOL file name

Cause: While using the SPOOL command, you specified the name of a file that already exists.

Action: Specify a unique valid SPOOL file name.

PGU-00101 extraneous text at end of command

Cause: There were unrecognized commands or other text on the command line.

Action: Check the syntax of the command, and then issue the command again.

PGU-00106 invalid ECHO switch

Cause: An invalid option for the SET ECHO command was specified.

Action: Use either ON or OFF as an option for the SET ECHO command.

PGU-00107 invalid TERMOUT switch

Cause: An invalid option for the SET TERMOUT command was specified.

Action: Use either ON or OFF as an option for the SET TERMOUT command.

PGU-00108 invalid TIMING switch

Cause: An invalid option for the SET TIMING command was specified.

Action: Use either ON or OFF as an option for the SET TIMING command.

PGU-00110 illegal SET option

Cause: An invalid option for the `SET` command was specified.

Action: Check the syntax of the `SET` command, and issue the command again.

PGU-00111 illegal SHOW option

Cause: An invalid option for the `SHOW` command was specified.

Action: Check the syntax of the `SHOW` command, and issue the command again.

PGU-00115 unexpected end of command

Cause: An option was specified without the required arguments.

Action: Check the syntax of the command, and enter the command again with the appropriate arguments for options that require values.

PGU-00120 invalid STOPONERROR switch

Cause: An invalid options switch was specified for the `SET STOPONERROR` command.

Action: Use either `ON` or `OFF` as an option for the `SET STOPONERROR` command.

PGU-00122 invalid SET numeric parameter

Cause: A character or an invalid value was specified when a number was expected as a value for a `SET` command option.

Action: Check the syntax of the command, use an appropriate number for the option, and enter the command again.

PGU-00125 integer value overflow

Cause: A numeric value was specified that was too large.

Action: Use a smaller number.

PGU-00129 value out of range (1 - max)

Cause: The specified value was out of range. The valid range is given by the error message.

Action: Use a number within the range specified by this error.

PGU-00132 null *hostname/password* specified

Cause: The `host name` or `password` was not specified.

Action: Specify a `host name` and `password`.

PGU-00136 bad variable specification

Cause: A variable was incorrectly specified by using the `VARIABLE` command.

Action: Check the syntax of the command and then issue the command again.

PGU-00137 syntax error in PL/SQL Block

Cause: The `PL/SQL` block contains a syntax error.

Action: Correct the syntax error.

PGU-00142 cannot recognize object type, owner or name

Cause: The specified object type, owner, or name was not recognized.

Action: Specify a legal object type, owner, or name.

PGU-00143 variable has not been defined

Cause: The specified variable was not recognized.

Action: Specify an existing variable.

PGU-00144 invalid object type for DESCRIBE

Cause: The specified object type was not TABLE, VIEW or PROCEDURE.

Action: Check whether the object is a table, view, or procedure. If yes, then check whether you specified the correct name and try again. If not, then you cannot describe the object.

PGU-00145 invalid object name for DESCRIBE

Cause: The specified table, view, stored procedure or function was not recognized.

Action: Check the spelling and ensure that you specify an existing table, view, stored procedure or function.

PGU-00149 invalid SERVEROUTPUT switch

Cause: An invalid option was specified for the SET SERVEROUTPUT command.

Action: Check the syntax of the command, and then issue the command again.

PGU-00300 internal error code; arguments: [arg1], [arg2]

Cause: You have encountered an internal error.

Action: Supported customers can call Oracle Support Services and tell them the circumstances that lead to the error with the complete set of received messages.

PGU-00302 not connected to a database

Cause: You were not connected to the database, as required for the requested operation.

Action: Connect to the database using a valid user name and password before retrying the operation.

PGU-00304 input file I/O error [errno] - input aborted

Cause: A command file that is used as input to PGAU was corrupt or invalid.

Action: Check the file before retrying the operation.

PGU-00305 command size exceeds internal buffer size (size)

Cause: The SQL statement size exceeded the buffer size of PGAU.

Action: Shorten the SQL statement by removing extra blanks or by using intermediate statements as views if necessary.

PGU-00306 monitor cycle interval time out of range (1 - max)

Cause: You entered an invalid number for the cycle interval.

Action: Enter a number between 1 and 3600 for the cycle interval. This number indicates seconds.

PGU-00307 cannot open spool file filename

Cause: PGAU tried to open a spool file after you entered SPOOL file name, but it could not open the file. Possible causes could be lack of disk space or inadequate privileges to create a file.

Action: Determine why PGAU could not create a new file, and then retry.

PGU-00308 no spool file opened

Cause: You entered SPOOL OFF, but you were not spooling, so there was no file to close.

Action: If you want to capture session output, first use the `SPOOL` command to open a file and then enter your commands before closing the file with `SPOOL OFF`.

PGU-00309 cannot close spool file *filename*

Cause: `SPOOL OFF` could not close the currently open spool file.

Action: Check with the operating system for the reason why the spool file could not be closed.

PGU-00310 cannot open parameter file *filename*

Cause: `PGAU` cannot locate or open the file specified by the `PFILE` option, either because the file does not exist, or because `PGAU` has insufficient privileges to open the file.

Action: Check that the file exists in a location expected by `PGAU` and that it can be opened.

PGU-00311 data exceeds internal buffer size

Cause: The results returned by a SQL query exceed the internal `PGAU` buffer.

Action: Use the `SET` command to increase `MAXDATA` or to decrease `ARRAYSIZE`.

PGU-00314 invalid parameter given on PGAU command line

Cause: An unrecognized parameter was given on the `PGAU` command line.

Action: Check the parameters given on the `PGAU` command line.

PGU-00315 cannot open command file *filename*

Cause: `PGAU` cannot locate the specified command file.

Action: Verify the file name, and check whether `PGAU` can access to it.

PGU-00317 version of tool conflicts with version *ver* of DATA

Cause: This version of `PGAU` cannot process the `DATA` in the database.

Action: You attempted to use a tool that might cause damage to the current database. If the version of `DATA` is greater than the version of the tool, then you are using the wrong version of the tool. If the version of the tool is greater than the version of the `DATA`, then check for an Oracle-supplied SQL script to update the `DATA` to the correct version level.

PGU-00318 PGAU command line error [*linenum*]

Cause: You made a syntax or typing error while entering a `PGAU` command line.

Action: Check the syntax and try again.

PGU-00319 cannot locate pgau configuration file, *filename*

Cause: The specified file `filename` cannot be located.

Action: Check that the specified file exists before rerunning `PGAU`.

PGU-00320 cannot open pgau configuration file, *filename*

Cause: The specified file `filename` cannot be opened.

Action: Check that the specified file is available for use by `PGAU`.

PGU-00322 total size of command line parameters exceeds buffer size

Cause: You entered too many command-line arguments and the `PGAU` buffer was exceeded.

Action: Reduce the number of command-line arguments.

PGU-00325 pfile too large

Cause: The file that you specified using `PFILE` is too large (exceeds 8 Kb).

Action: Reduce the size of the parameter file before specifying it with `PFILE`.

PGU-00327 command not available in this mode

Cause: You specified a command that is not available in this mode.

Action: Do not specify the command.

PGU-00328 insufficient privilege for this display

Cause: You attempted to display a `MONITOR` display without sufficient privileges.

Action: Contact the DBA to obtain the required privileges.

PGU-00331 cannot allocate enough memory for SQL Buffer

Cause: There is not enough memory for the current SQL buffer.

Action: Use the `SET` command to reduce `MAXDATA`.

PGU-00337 missing instance name

Cause: The instance name was not specified in the connect statement.

Action: Use `connect username/password@instance` or `connect username/password`.

PGU-00341 'name' is an undefined bind variable

Cause: The SQL statement refers to an undefined bind variable.

Action: Use the `VARIABLE` statement to define the bind variable and rerun the query.

PGU-00347 no offline tablespaces exist

Cause: The list box of set tablespace online contained no elements.

Action: No action is required.

PGU-00359 monitor already active

Cause: An instantiation of this monitor is already active.

Action: Cycle through windows until this monitor becomes visible.

PGU-00360 object to be described does not exist

Cause: The object in a `DESCRIBE FUNCTION, PROCEDURE, PACKAGE` statement does not exist.

Action: Check that the object name and owner are correct and that the object exists.

PGU-00361 error during describe

Cause: An unexpected error occurred during a describe.

Action: Check the following error, and correct the problem.

PGU-00362 object name is a package; use DESCRIBE package.procedure

Cause: The named object is a package. `DESCRIBE` does not currently describe an entire package specification.

Action: Describe the package specification as indicated in the message text.

PGU-00363 procedure or function name not found in the package

Cause: The named package does not contain the procedure or function specified.

Action: Specify a procedure or function within the package.

PGU-00364 object *name* is a remote object, cannot further describe

Cause: The specified object name contains a database link, or the object is a synonym that resolves to a name with a database link. Such objects cannot currently be described.

Action: Specify a local object.

PGU-00365 object *name* is invalid, it may not be described

Cause: The object must have been successfully compiled.

Action: Fix any errors in the object and recompile.

PGU-00366 name *name* is malformed, it must be of form `[[a.]b.]c@dblink`

Cause: The name cant have at most three parts and a `dblink`.

Action: Use a well-formed object name.

PGU-00370 mandatory field/list needs to be filled in

Cause: You tried to execute the dialog before filling in all the required items.

Action: Fill in all the required items and retry.

PGU-00371 cannot open/locate input help file, *filename*

Cause: You pressed the Help key or chose an item from the Help menu.

Action: Save the help file in the location specified.

PGU-00372 cannot open/locate input index file, *filename*

Cause: You pressed the Help key or chose an item from the Help menu.

Action: Save the index help file in the location specified.

PGU-00373 cannot allocate memory of size *size* from toolkit

Cause: You pressed the Help key or chose an item from the Help menu.

Action: Exit PGAU, reenter and then try again.

PGU-00374 could not set file position in *filename* directly

Cause: You pressed the Help key or chose an item from the Help menu.

Action: The `fseek` routine failed. The Alternate is chosen and no action is necessary.

PGU-00375 unexpected end of file, *filename*

Cause: You pressed the Help key or chose an item from the Help menu.

Action: Exit PGAU, and install the correct help file.

PGU-00376 mandatory field has to be filled in before navigation is possible

Cause: You attempted to navigate to the next or previous item without filling in a mandatory field.

Action: Fill in the mandatory field and then go to the next or previous item.

PGU-00377 mandatory list item has to be selected before navigation is possible

Cause: You attempted to navigate to the next or previous item without selecting a mandatory item.

Action: Select an item and then go to the next or previous item.

PGU-00378 an item has to be selected before help can be sought for it

Cause: An attempt was made to obtain help before selecting an item.

Action: Select an item first and then try to obtain help.

PGU-00379 variable(s) not defined

Cause: An attempt was made to access or show one more more variables (SHOW VAR [TABLES] [var-name]), were not specified.

Action: Ensure that the variables are correctly defined.

PGU-00380 procedural option required for this statement

Cause: A statement that requires a procedural option was run, but the procedural option is not installed.

Action: This statement cannot be issued without the corresponding procedural option. Install the procedural option to run this statement.

PGU-00381 error in the SERVEROUTPUT option

Cause: The most likely cause is that the DBMS_OUTPUT package is not installed. Check the accompanying messages for more information.

Action: Check accompanying messages and take appropriate action.

PGU-00382 value *filename* is not a recognizable file name for *variable*

Cause: The value for the PGAU initialization variable that points to the initialization file is not a recognizable file name.

Action: Specify a valid file name or do not define the variable to avoid running the initialization file.

PGU-00383 file name *filename* pointed to by *variable* could not be opened

Cause: The file name pointed to by the PGAU initialization variable could not be opened.

Action: Specify an available file or do not define the variable to avoid running the initialization file.

PGU-00384 could not open pgau resource file, *filename*

Cause: The specified file could not be opened.

Action: Make the file available to PGAU.

PGU-00385 could not locate pgau resource file, *filename*

Cause: The specified file could not be located.

Action: Ensure that the file exists before rerunning PGAU.

PGU-00386 could not open toolkit resource file, *filename*

Cause: The specified file could not be opened.

Action: Make the file available to PGAU.

PGU-00387 could not locate toolkit resource file, *filename*

Cause: The specified file could not be located.

Action: Ensure that the file exists before rerunning PGAU.

PGU-00388 cannot start PGAU in screen mode; check if values are legal

Cause: A variable was not properly set.

Action: Check the values listed by PGAU to see if they are correct.

PGU-00389 toolkit resource file name is defined to *filename*

Cause: The file `filename` could not be located.

Action: Ensure that the specified file exists and can be accessed.

PGU-00390 terminal type *term* is defined by *variable*

Cause: The terminal type specified by `variable` might not be valid.

Action: Ensure that the terminal specification is correct.

PGU-00391 value *var* is defined to *val*; legal values: *val* or *val*

Cause: The PGAU mode has been incorrectly defined.

Action: Define the value of the variable as one of the legal values.

PGU-00501 through PGU-00701

Cause: These are informational messages.

Action: No action is required.

PGU-20000 *oper* of *statement* statement failed for table *table*, *rc=rc*

Cause: PGAU encountered a syntax error during the parse operation (`oper`) of the statement (`statement`) for the table (`table`) with the indicated return code (`rc`).

Action: Check the PGAU statement identifier names for proper spelling and punctuation or other syntax errors.

PGU-20001 *oper* of *varname* variable of *statement* statement failed for table *table*, *rc=rc*

Cause: A semantic error has occurred. The operation (`oper`) on the variable (`varname`) for the statement (`statement`) against the PG DD table (`table`) failed with return code (`rc`).

Action: Check all the identifier names in the PGAU script.

PGU-20002 unable to login to Oracle.

Cause: An error occurred while attempting to log on to the Oracle server. Only syntax checking continues with the next statement.

Action: Ensure that the Oracle server containing the PG DD is operational.

PGU-20003 unable to open the cursor.

Cause: This is an internal PGAU logic error that should not occur. An error occurred opening a SQL statement cursor. Only syntax checking continues with the next statement.

Action: Ensure that the Oracle server containing the PG DD is operational. Otherwise, regenerate the error with symptom documentation. Supported customers should contact Oracle Support Services for assistance.

PGU-20004 unable to close the cursor

Cause: This is an internal PGAU logic error that should not occur. An error occurred closing a SQL statement cursor. Only syntax checking continues with the next statement.

Action: Ensure that the Oracle server containing the PG DD is operational. Otherwise, regenerate the error with symptom documentation. Supported customers should contact Oracle Support Services for assistance.

PGU-20005 unable to commit PG DD changes

Cause: This is an internal PGAU logic error that should not occur. An error occurred, while attempting to commit changes to the PG DD. Only syntax checking continues with the next statement.

Action: Ensure that the Oracle server containing the PG DD is operational. Otherwise, regenerate the error with symptom documentation. Supported customers should contact Oracle Support Services for assistance.

PGU-20006 unable to logout from Oracle

Cause: This is an internal PGAU logic error that should not occur. An error occurred during logoff from the Oracle server. Only syntax checking continues with the next statement.

Action: Ensure that the Oracle server containing the PG DD is operational. Otherwise, regenerate the error with symptom documentation. Supported customers should contact Oracle Support Services for assistance.

PGU-20007 unable to rollback PG DD changes

Cause: This is an internal PGAU logic error that should not occur. An error occurred attempting to roll back changes from the PG DD. Only syntax checking continues with the next statement.

Action: Ensure that the Oracle server containing the PG DD is operational. Otherwise, regenerate the error with symptom documentation. Supported customers should contact Oracle Support Services for assistance.

PGU-30000 transaction *tranname*, version *version* specification generated to file *fileid*

Cause: PGAU successfully completed generation of the TIP specification for the transaction (*tranname*) and the version (*version*) into the file (*fileid*).

Action: No action is required. This is an informational message.

PGU-30001 PGAU internal error - *module*

Cause: An internal PGAU operation error has occurred.

Action: Supported customers should contact Oracle Support Services for assistance.

PGU-30002 >>> Initialization of *type* File Header <<<

Cause: This record is written to the trace or log file upon initialization.

Action: No action is required. This is an informational message only.

PGU-30003 transaction *tranname*, version *version* body generated to file *fileid*

Cause: PGAU successfully completed generation of the TIP body for the transaction (*tranname*) and the version (*version*) into the file (*fileid*).

Action: No action is required. This is an informational message.

PGU-30008 failure to open file *fileid*

Cause: Before the complete initialization of LMS functions, PGAU tracing or logging to disks was requested, but PGAU failed to open the indicated trace or log file (*fileid*).

Action: Ensure that write access is available for the file and that memory space for file growth is available.

PGU-30009 failure to close file *fileid*

Cause: After termination of LMS functions, PGAU tracing or logging to disks was requested, but PGAU failed to close the indicated trace or log file (*fileid*).

Action: Ensure that write access is available for the file and that memory space for file growth is available.

PGU-30010 failure to obtain *bytes* bytes of storage for *use*

Cause: Before the complete initialization of LMS functions, PGAU failed to obtain *bytes* of storage for *use* that meets the requirements specified in the with LMS initialization.

Action: Increase the total amount of private storage available to PGAU during operation.

PGU-30012 invalid language environment (*lang*), default used

Cause: Before complete initialization of LMS functions, the environment variable LANGUAGE specified an invalid Oracle NLS language, *lang*. The language environment variable (SSTDLANG) setting was used as the default.

Action: Correct the LANGUAGE environment variable to specify a valid Oracle NLS language.

PGU-30014 pgau message file *fileid* not found

Cause: Before complete initialization of LMS functions, the PGAU message file (*fileid*) could not be located.

Action: Ensure that the Oracle message file resides in \$ORACLE_HOME/pg4appc/mesg directory and is not damaged. Valid file IDs are of the form *pguxx.msb*, where *xx* is the country code.

PGU-30024 memory allocate failure: *purpose* for *len* bytes

Cause: Insufficient memory. Storage was not allocated for the reason *purpose*, of the specified size, *len*.

Action: Increase the total amount of private storage available to PGAU during the GENERATE operation.

PGU-30026 open failure: file *file* in *mode* mode

Cause: There was an error during *fopen* while processing *file* for *mode* access.

Action: Ensure that PGAU has access to the file specified and write access if the file is used for output.

PGU-30028 file *oper* failure: *fileid* rc *rc*

Cause: An error occurred during *oper* processing for the file, *fileid*, with return code, *rc*. File operations include *fclose* to close input or output files, or *remove* to erase, delete, or remove output files.

Action: Ensure that PGAU has access to the file specified and write access if the file is used for output.

Messages PGU-30030 to PGU-30637

PGU-30030 read failure: *oper* from *file* after offset *offset*

Cause: There was an error during *fgets* *oper* processing for *file* at *offset* bytes into the file.

Action: Ensure that PGAU has access to the file specified.

PGU-30031 write failure: *oper* to *fileid* at *addr* for *len* bytes

Cause: There was an error during *fputs* *oper* processing for *file* from buffer at *addr* for *len* bytes of data.

Action: Ensure that PGAU has access to the file specified and that disk space is available for file growth.

PGU-30032 write log failure: rc rc, message no was msgno

Cause: The `fprintf` routine returned the indicated error code `rc` when attempting to print PGAU error message number `msgno`. The indicated event message was being written when the error occurred.

Action: Ensure that PGAU has write access to the `pgau.log` file and that disk space is available for file growth.

PGU-30033 write trace failure: rc rc, message no was msgno

Cause: The `fprintf` routine returned the indicated error code `rc` when attempting to print PGAU trace message number `msgno`. The indicated trace message was being written when the error occurred.

Action: Check to ensure that PGAU has write access to the `pgau.trc` file and that disk space is available for file growth.

PGU-30035 file control failure: oper for file, rc rc

Cause: There was an error during control function `oper` for `file` and error code `rc` was returned.

Action: Ensure that PGAU has access to the file specified and that disk space is available for file growth.

PGU-30036 no ORACLE_HOME environment variable available

Cause: The `ORACLE_HOME` environment variable setting is missing or invalid.

Action: Correct the `ORACLE_HOME` environment variable to specify the current Oracle home directory.

PGU-30120 missing attr attribute for parm parmname, field field, f# f#

Cause: The `parmname` parameter specified an aggregate data record for which the field `field` with PG DD field ID number `f#` was being processed, but the field had no relative level attribute `attr`. Typical relative level attributes is:

`nn` COBOL-clauses for IBMVSCOBOLII 'LEVEL' attribute

Each attribute is stored as a separate row in the PG DD, and the row for the missing attribute might have been deleted from the PG DD.

Action: Check the PG DD to ensure that the failing field has a level attribute assigned.

PGU-30122 level limit: field f#=f# nest level count exceeds min/max of limit

Cause: PGAU was processing nested records when the next field `f#` for nest level count exceeded the defined minimum or maximum nest level `limit` for the product.

Action: Simplify the data record definition to reduce nest levels.

PGU-30123 level limit: level levN field fld# can't follow level levC

Cause: PGAU was processing nested records when the next field at level `levN`, identified by `fld#`, occurred after fields at current level `levC`. Additional fields at intermediate levels should have preceded this field. This error occurs when special fields such as COBOL RENAME (at LEVEL 66), appear next after the level 01 definition.

Action: Correct the data record definition and place the special fields in their proper sequence, or insert fields being modified ahead of the special modifying field.

PGU-30220 allocate failure: *type* storage of *num* bytes

Cause: Insufficient memory was available for the specified purpose. The requestor attempted to allocate a control block type of the specified size during normal operation.

Action: Increase the total amount of private storage that is available to PGAU during the GENERATE operation.

PGU-30233 control block load failure: *type* block from PG DD

Cause: A control block of the specified type was being loaded with a row from the PG Data Dictionary when an error occurred. A preceding message was issued specifying the cause.

Action: Refer to the recommended action for the preceding message.

PGU-30234 attribute array for *type* exceeded by PG DD rows > *limit*

Cause: A control block of the specified type was being loaded with attribute rows from the PG Data Dictionary when the limit was exceeded.

Action: Regenerate the error with tracing enabled and determine which PG DD rows are exceeding the attribute limit. Delete any possible duplicates. Supported customers should contact Oracle Support Services for assistance.

PGU-30300 work file output error writing *fileid*

Cause: PGAU encountered an error while writing a PL/SQL code segment to the indicated work file *fileid*.

Action: Ensure that PGAU has access to the file specified and that memory space is available for file growth.

PGU-30301 TIP output error appending *infile* to *outfile*

Cause: PGAU encountered an error appending the work file, *infile*, to the TIP output file, *outfile*.

Action: Ensure that PGAU has access to both files specified and that disk space is available for TIP output file growth.

PGU-30302 TIP output error writing function *func* to file *file*

Cause: PGAU encountered an error appending the indicated TIP function definition (*func*) to the work output file (*file*). An error occurred during generation of the TIP Package Specification.

Action: Ensure that PGAU has write access to the file specified and that memory space is available for TIP output work file growth.

PGU-30303 TIP generation error writing variable *var* to file *file*

Cause: PGAU encountered an error appending the TIP variable definition *var* to the work output file (*file*). An error occurred during generation of the TIP Package Specification.

Action: Ensure that PGAU has write access to the file specified and that disk space is available for TIP output work file growth.

PGU-30304 TIP generation warning: *field* name truncated to *len* characters

Cause: This is a warning of possible non-unique PL/SQL names. PGAU encountered record field names that, when combined, exceed the maximum

PL/SQL name length. The last field name specified (*field*) was truncated to the length *len*. This condition occurred during generation of the TIP Package Specification for parameter variables.

Action: Revise the PG DD entries for the defined parameters, and either reduce the number of nested record levels, or shorten the record field names.

PGU-30305 TIP generation error defining record type *type* for name

Cause: An error occurred during the generation of a nested record type (*type*) for the indicated field name (*name*). This message should be preceded by the specific error. This condition occurred during the generation of the TIP Package Specification for parameter variables.

Action: Follow the recommended action for the first error message issued.

PGU-30306 TIP generation: invalid PL/SQL parameter mode *mode* for parm

Cause: The parameter call mode (*mode*) specified for the parameter (*parm*). Valid parameter modes are IN, OUT, or IN OUT. This is a probable PG DD content error.

Action: Revise the PG DD entries for the indicated parameter, and correct the PL/SQL parameter call mode for the indicated parameter.

PGU-30307 TIP generation: invalid PL/SQL variable type *type* for field

Cause: An invalid PL/SQL record type (*type*) was encountered while generating conversion statements for a data field (*field*). The indicated type is neither valid PL/SQL nor a nested record type for the indicated record field. This is a probable PG DD content error.

Action: Revise the PG DD entries for the indicated parameter, and either correct the PL/SQL variable type for the field, or define a corresponding nested record for the field.

PGU-30308 TIP generation: name *recname.varname* exceeds max length of *maxlen* characters

Cause: The number of nested groups and lengths of their field names in the input data, when concatenated to form a fullyqualified PL/SQL record field variable name (*recname.varname*), exceeded the maximum length allowed by PL/SQL, *maxlen*. This error occurred during generation of the TIP package specification for parameter variables.

Action: Revise the PG data dictionary entries for the defined parameters, and either reduce the number of nested record levels, or shorten the record field names.

PGU-30309 TIP generation: call *cname parm parm* exceeds PGAXFER type limit (*limit*)

Cause: The TIP function call, *cname* at parameter, *parm*, exchanges too many parameters for the *type* buffers on a PGAXFER RPC. The PGAXFER parameter limit is *limit*. The indicated *type* is either SEND or RECEIVE. This is probably a PGAU DEFINE CALL error, coding too many parameters. The IN OUT mode parameters are present in both send and receive buffers and might be causing the limit problem.

Action: Revise the transaction and call to specify the excessive parameters on an additional function call, and include the added call in the transaction definition.

PGU-30314 TIP generation: failure converting record type *type* for name.

Cause: An error occurred during the generation of PL/SQL statements for conversion of a nested record type (*type*) for the indicated field name (*name*).

This message should be preceded by the specific error. The condition occurred during the generation of the TIP Package Specification for parameter variables.

Action: Follow the recommended action for the first error message issued.

PGU-30315 TIP generation: parameter conversion error for *dname*, *call*

Cause: The PL/SQL conversion statements for the indicated parameter data (*dname*) for function *call*, could not be generated. This message might be preceded by specific messages describing the error. Possible errors include disk access, invalid datatype and missing nested record or field datatype definitions.

Action: Follow the recommended action for the first error message issued, and ensure that the PG DD entries for the indicated parameter are correct and that disk access is allowed to output work files.

PGU-30316 TIP generation: PL/SQL *stmt* control logic for *fname clause*

Cause: The PL/SQL statement, *stmt*, that is used to control TIP processing of the field, *fname* for compiler clause, *clause*, could not be generated.

Typical IBMVSCOBOLII clauses are:

OCCURS - requires FOR/LOOP logic

OCCURS DEPENDING ON - requires FOR/LOOP logic

REDEFINES - requires IF/END IF logic

This message might be preceded by specific messages describing the error. Possible errors include disk access, invalid datatype, missing nested record or field datatype definitions.

Action: Follow the recommended action for the first error message issued and ensure that the PG DD entries for the indicated parameter are correct and that disk access is allowed to output work files.

PGU-30317 TIP generation: PL/SQL *stmt* statements for *dname cname*

Cause: The attempt to generate PL/SQL control logic statements (*stmt*) to process the TIP parameter *dname* in the function *cname* failed. This message might follow more specific messages describing the error. Possible errors include disk access, invalid datatype, and missing next record or field datatype definitions.

Action: Follow the recommended action for the first error message issued, and ensure that the PG DD entries for the indicated parameter are correct and that disk access is allowed to the output work files.

PGU-30318 TIP generation: *d-field f#*, PL/SQL *stmt* for *num*

Cause: Data field, *d-field f#*, was referenced by other fields, which required that PL/SQL statements (*stmt*) be generated for the value *num*, but an error occurred during the generation of the statements. This message might follow specific messages describing the error. Possible errors include disk access and memory exceeded.

Action: Follow recommended action for the first error message issued and ensure that disk access is allowed to output work files.

PGU-30319 TIP generation: *type id#*

Cause: TIP execution diagnostics for PG DD references PKGEX (DR) was requested on the PGAU GENERATE statement, but the PG DD reference for *type id#* could not be generated. This message might be preceded by specific messages describing the error. Possible errors include disk access and memory exceeded.

Action: Follow recommended action for the first error message issued, and ensure that disk access is allowed to output work files.

PGU-30600 TIP generation: invalid COBOL syntax in *field*

Cause: The COBOL data field (*field*) specifies conflicting or invalid PIC *mask* and USAGE *maskopts* clauses. The invalid PIC and USAGE definitions were selected from the PG DD to identify field conversion function calls. PGAU also issues messages 30601 for *mask* and 30602 for *maskopts*.

Action: Use the `pgddsfc.sql` script to regenerate the selected rows and then correct the *mask* and *maskopts* columns for the stated field in the PG DD.

PGU-30601 COBOL mask: *mask*

Cause: Issued to identify *mask* for the preceding message.

Action: See preceding message.

PGU-30602 COBOL maskopts: *maskopts*

Cause: Issued to identify *maskopts* for the preceding message.

Action: See preceding message.

PGU-30603 TIP generation: *field syntax*

Cause: See message PGU-30604.

Action: Refer to message PGU-30604 for more information.

PGU-30604 missing *attr* attribute for *token*

Cause: A COBOL field, *field*, containing a clause of the form *syntax* required a missing attribute, *attr*, corresponding to *token*. PGAU issues messages PGU-30603 and PGU-30604 consecutively for *field*, *syntax*, *attr*, and *token*.

Typical IBMVSCOBOLII required token attributes are:

- *field* OCCURS int-1 TIMES
where int-1 is the token for attribute `repprpff`
- *field* OCCURS int-1TOint-2 TIMES DEPENDING ON name-1
where:
int-1 is the token for attribute `repprpvf`
int-2 is the token for attribute `repprpv1`
name-1 is the token for attribute `repprpvm`
- *field* RENAMES name-1 THRU name-2
where:
name-1 is the token for attribute `renamemf`
name-2 is the token for attribute `renamem1`
- *field* REDEFINES name-1 WHEN name-2 = value
where:
name-1 is the token for attribute `remapsmf`
name-2 is the token for attribute `remapsm1`
value is the token for attribute `remapswc` or `remapswn`
- *field* LENGTH IS name-1

where:

`name-1` is the token for attribute length

Each attribute is stored as a separate row in the PG DD, and the missing attributes row might have been deleted from the PG DD.

Action: In the PG DD, redefine the referencing data item to restore the missing attribute row.

PGU-30605 TIP generation: invalid COBOL syntax in unknown field:

Cause: An unidentified COBOL field specifies conflicting or invalid PIC mask and USAGE maskopts clauses. The invalid PIC and USAGE definitions were selected from the PG DD to identify global TIP conversion variables. PGAU also issues messages 30601 for mask and 30602 for maskopts.

Action: Use the `pgddsxf.sql` script to regenerate the selected rows and then correct the mask and maskopts columns for the invalid row in the PG DD.

PGU-30606 TIP generation: missing multibyte NLS name for field

Cause: The COBOL data field (`field`) specifies character data in the PIC mask and USAGE maskopts clauses that require translation with a multibyte character set (MBCS), but no Oracle MBCS NLS name was specified. The MBCS-oriented PIC and USAGE definitions were selected from the PG DD to identify field conversion function calls. PGAU also issues messages PGU-30601 for mask and PGU-30602 for maskopts.

Action: Use the `pgddsxa.sql` script to regenerate the selected rows, and then either correct the mask and maskopts columns for the stated field in the PG DD to not require MBCS translation, or use the PGAU REDEFINE DATA FIELD NLS_LANGUAGE (`nlsname`) parameter to specify an Oracle MBCS NLS name for the field, or specify the NLS_MBCS (`nlsname`) parameter on the PGAU DEFINE TRANSACTION statement.

PGU-30610 TIP generation: d-field syntax

Cause: See message PGU-30611.

Action: Refer to message PGU-30611 for more information.

PGU-30611 references missing field r-field specified for token

Cause: The data field, `d-field`, containing a clause of the form `syntax`, references a missing field, `r-field`, specified by the clause word, `token`. PGAU issues messages PGU-30610 and PGU-30611 consecutively for `d-field`, `syntax`, `r-field`, and `token`.

Typical IBMVSCOBOLII clause syntax for this error includes:

```
d-field OCCURS int-1 TIMES
d-field OCCURS int-1 TO int-2 TIMES DEPENDING ON name-1
d-field RENAMES name-1 THRU name-2
d-field REDEFINES name-1 WHEN name-2 = value
d-field LENGTH IS name-1
```

The clause references a field, `r-field`, specified by the word `token`.

PGAU GENERATE searched previous fields within the current parameter but failed to find the named field, `r-field`. The word, `token` might be misspelled in the defining clause, or the referenced field whose name matches `token` might be missing from, or misspelled in, the PG DD.

Action: In the PG DD, redefine the entry for `d-field` to change its clause token to specify the correct name of the intended field, `r-field`. You can also redefine the data name of the intended field, `r-field`, to match the subsequent clauses token reference. Also, ensure that the data field, `d-field`, containing the clause follows the intended field, `r-field`, in the data definition.

PGU-30622 unable to insert *attr* value for field *d-field*

Cause: During TIP generation, a data field `d-field` was encountered for which no `attr` attribute existed. A new attribute row was created but could not be inserted into the PG DD.

Action: Check for an Oracle error message preceding this message to determine the cause of the problem.

PGU-30630 TIP generation: *d-field f#*

Cause: See message PGU-30631.

Action: Refer to message PGU-30631 for more information.

PGU-30631 no alignment for environment *d-field f#,datatype dtype dagno*

Cause: The data field `d-field f#` specified an attribute that specifies remote host boundary alignment. The PG DD environment tables were searched for alignment information for environment (`ename`) compiler (`compno`), data type (`dtype`), data type alignment group `dagno`, but no entries were found matching these characteristics. PGAU issues messages PGU-30360 and PGU-30631 for `d-field f#` and `ename compno dtype dagno`. PGAU requires the alignment information to properly align the remote host data in the TIP transfer buffers.

Supported environment and compilers are IBM370 for IBMVSCOBOLII compiler.

Supported alignment attributes are:

```
d-field ... PIC S9(n) dtype SYNC
d-field ... PIC S9(n) dtype SYNCHRONIZE
```

Supported datatype values are:

```
COMP
COMPUTATIONAL
COMP-4
COMPUTATIONAL-4
```

Datatype grouped under `dagno` by length and alignment are:

```
dagno = 1 for 2-byte length aligned on 2-byte boundary
dagno = 2 for 4-byte length aligned on 4-byte boundary
```

Possible causes are:

- The TRANSACTION entry in the PG DD for the TIP that was being generated might have specified an invalid ENVIRONMENT.
- The DATA entry in the PG DD was being referenced in the TIP function call that was generated might have specified an invalid LANGUAGE.
- The field data type and length within the DATA entry might be invalid or unsupported.
- The PG DD might be unavailable. Preceding messages indicate problems accessing the PG DD.

Action: In the PG DD, redefine the TRANSACTION or DATA entries to correct the specification of the ENVIRONMENT, LANGUAGE, or field data type and attributes. Ensure that the Oracle server that supports the PG DD is active. If the error

persists, then regenerate the error with tracing enabled. Supported customers can contact Oracle Support Services for assistance.

PGU-30632 TIP generation: *d-field f#, clause ignored for dtype*

Cause: The `d-field f#` data field specified a modifying clause (`clause`) that is ignored for `dtype` data type by the compiler in the remote host environment.

Supported environment and compiler is IBM370 for the IBMVSCOBOLII compiler.

`SYNC/SYNCHRONIZE` is ignored when `dtype` is one of the following:

`COMP-3/COMPUTATIONAL-3`
`DISP/DISPLAY`

Action: This is a warning message and no action is required. PGAU continues executing. You might want to ensure that the resulting TIP data conversion is consistent with the remote host data format.

PGU-30635 TIP generation: *d-field f#, nested repeating group for clause*

Cause: The `d-field f#` data field specified a repeating group clause (`clause`) while a previous repeating group is active. The repeating group in `d-field` cannot be nested within a previous repeating group. TIPs use `PL/SQL` tables to implement repeating groups, and `PL/SQL` tables are limited to a single key or subscript that cannot support nested repeating groups (a table of tables).

Action: Redefine the data so that the previous repeating group ends before another repeating group begins.

PGU-30636 TIP generation: *d-field f#, attr attribute value (attrval) ignored*

Cause: The `d-field f#` data field specified an attribute (`attr`) and its value, `attrval`, is ignored.

Supported environment and compiler is IBM370 for the IBMVSCOBOLII compiler.

The `LEFT attrval` is ignored when `attr` is one of the following:

`JUST/JUSTIFY`
`SYNC/SYNCHRONIZE`

Action: This is a warning message and no action is required. PGAU continues executing. You might want to ensure that the resulting TIP data conversion is consistent with the remote host data format.

PGU-30637 TIP generation: *d-field f#, attr attribute value (attrval) invalid*

Cause: The `d-field f#` data field specified an attribute `attr` whose value `attrval` is invalid or unsupported.

Action: In the `PG DD`, redefine the `FIELD` entry to correct the attribute clause to specify a supported value.

Messages PGU-35002 to PGU-39999

PGU-35002 failure to open cursor for statement *stmtname: rc rc*

Cause: An attempt to open an Oracle cursor for the `PGAU SQL` statement, `stmtname`, failed with the Oracle error `rc`. This message is preceded by an Oracle server message for the specific error encountered.

Action: Ensure that the Oracle server that is supports the `PG DD` is active. If the error persists, then regenerate the error with tracing enabled. Supported customers should contact Oracle Support Services for assistance.

PGU-35003 failure to parse SQL statement *stmtname* for cursor *curno*: rc rc

Cause: An attempt to parse the PGAU SQL statement, *stmtname*, for Oracle cursor, *curno*, failed with Oracle error *rc*. This message is preceded by an Oracle server message for the specific error encountered.

Action: Ensure that the Oracle server that supports the PG DD is active. If the error persists, then regenerate the error with tracing enabled. Supported customers should contact Oracle Support Services for assistance.

PGU-35004 failure to bind *invar* for cursor *curno* statement *stmtname*: rc rc

Cause: An attempt to bind an input variable (*invar*) for Oracle cursor, *curno*, to the PGAU SQL statement, *stmtname*, failed with Oracle error *rc*. This message is preceded by an Oracle server message for the specific error encountered.

Action: Ensure that the Oracle server that supports PG DD is active. If the error persists, then regenerate the error with tracing enabled. Supported customers should contact Oracle Support Services for assistance.

PGU-35005 failure to define *outvar* for cursor *curno* statement *stmtname*: rc rc

Cause: An attempt to define the variable, *outvar*, for Oracle cursor, *curno*, to the PGAU SQL statement, *stmtname*, failed with Oracle error *rc*. This message is preceded by an Oracle server message for the specific error encountered.

Action: Ensure that the Oracle server that supports the PG DD is active. If the error persists, then regenerate the error with tracing enabled. Supported customers should contact Oracle Support Services for assistance.

PGU-35006 failure to execute for cursor *curno* statement *stmtname*: rc rc

Cause: The Execution of Oracle cursor, *curno*, to the PGAU SQL statement, *stmtname*, failed with Oracle error *rc*. This is possible error in a PGAU GENERATE statement transaction or version parameters, the missing rows, or misspelling in the PG DD for the requested transaction. This message is preceded by an Oracle server message for the specific error encountered.

Action: Check that all call, data, and attribute definitions associated with the requested transaction and version are properly defined in the PG DD. If the error persists, then regenerate the error with tracing enabled. Supported customers should contact Oracle Support Services for assistance.

PGU-35007 failure to fetch for cursor *curno* statement *statement*: rc rc

Cause: A fetch operation using Oracle cursor, *curno*, to the PGAU SQL statement, *stmtname*, failed with Oracle error *rc*. This is possible error in a PGAU GENERATE statement transaction or version parameters, the missing rows, or misspelling in the PG DD for the requested transaction. This message is preceded by an Oracle server message for the specific error encountered.

Action: Check that all call, data, and attribute definitions associated with the requested transaction and version are properly defined in the PG DD. If the error persists, then regenerate the error with tracing enabled. Supported customers should contact Oracle Support Services for assistance.

PGU-35008 failure to close cursor *curno* for statement *stmtname*: rc rc

Cause: A close operation of the Oracle cursor, *curno*, for PGAU SQL statement, *stmtname*, failed with Oracle error *rc*. This message is preceded by an Oracle server message for the specific error encountered.

Action: Ensure that the Oracle server that supports the PG DD is active. If the error persists, then regenerate the error with tracing enabled. Supported customers should contact Oracle Support Services for assistance.

PGU-35009 no transaction rows for statement *statement*, *tname tver:rc rc*

Cause: Transaction rows were not fetched from the PG DD for the PGAU SQL statement, *stmtname*, with the transaction name, *tname*, and transaction version *tver*. The Oracle error code is *rc*. Either the transaction name and version supplied on the GENERATE statement were invalid, or the transaction entry was missing from the PG DD. This message is preceded by an Oracle server message for the specific error encountered.

Action: Ensure that the requested transaction and version are properly defined in the PG DD, or correct the GENERATE statement.

PGU-35010 no *type* rows for statement *statement*, *id no idno: rc rc*

Cause: The *type* rows were not fetched from the PG DD for the PGAU SQL statement, *statement*, with ID number, *idno*. The Oracle error code is *rc*. Either the parent PG DD entry references an invalid subordinate entry or the entry is missing from the PG DD. This message is preceded by an Oracle server message for the specific error encountered.

Action: Ensure that the requested transaction and version are properly defined in the PG DD, or correct the GENERATE statement.

PGU-35011 no environment rows for statement *statement*, *ename compno: rc rc*

Cause: The environment rows were not fetched from the PG DD for the PGAU SQL statement, *statement*, with environment name *ename* and compiler number *compno*. The Oracle error code is *rc*. Either the environment specified for the transaction or the compiler specified for the data entry in the transaction were invalid, or the entries were missing from the PG DD. This message is preceded by an Oracle server message for the specific error encountered.

Action: Ensure that the requested transaction and version are properly defined in the PG DD, or correct the GENERATE statement.

PGU-35012 missing transaction entry *tname tver*

Cause: The transaction *tname tver* was not found in the PG DD. Either the transaction name and version supplied on the GENERATE statement were invalid, or the transaction specified in a previous DEFINE TRANSACTION, *tname*, might have been deleted from, or altered in, the PG DD. This message might be preceded by specific messages describing the error.

Action: Correct the requested transaction and version definitions in the PG DD, or correct the GENERATE statement.

PGU-35013 missing call entry for transaction *tname tver (tin)*

Cause: A call entry under the transaction definition *tname tver (tin)*, was not found in the PG DD. The call entry that is associated with the transaction (as specified in a previous DEFINE TRANSACTION ... CALL(*cname*)) might have been deleted from, or altered in, the PG DD. This message might be preceded by specific messages describing the error.

Action: Correct the requested transaction or call definitions in the PG DD, or correct the GENERATE statement to request a different transaction.

PGU-35014 missing parm for transaction *tname tver (tin)* call *cname cver (cin)*

Cause: A parameter or data entry for the transaction *tname tver (tin)*, call *cname cver (cin)*, was not found in the PG DD. The *parm* or data entry associated with the call (as specified in a previous DEFINE CALL *cname* ... PARMS(*dname*)) might have been deleted from, or altered in, the PG DD. This message might be preceded by specific messages describing the error.

Action: Correct the requested transaction or call definitions in the PG DD, or correct the GENERATE statement to request a different transaction.

PGU-35015 missing field entry for transaction *tname tver (tin)* call *cname cver (cin)*, parm *relno dname dver (din)*

Cause: A data field entry for the transaction *tname tversion (tin)*, call *cname cversion (cin)* parameter *relno* and data *dname dversion (din)* was not found in the PG DD. The field entry associated with this data (as specified in the input file field position or FIELD (*fname*) of a previous DEFINE or REDEFINE DATA statement) might have been deleted from, or altered in, the PG DD. This message might be preceded by specific messages describing the error.

Action: Correct the requested transaction or call definitions in the PG DD, or correct the GENERATE statement to request a different transaction.

PGU-35016 missing formatted conversion entry for transaction *tname tver*

Cause: For transaction, *tname tver*, the data field usage, mask, and maskopts entries were not found in the PG DD. All PG DD tables are searched, starting with `pga_trans` through `pga_fields`, to select rows having `usage=PASS` and not null `mask` and `maskopts` columns. These are used to identify data fields use UTL_PG MAKE_..._FORMAT conversions.

Possible invalid entries in `pga_fields` (`usage`, `mask`, `maskopts`) columns. This message might be preceded by specific messages describing the error.

Action: Correct the data definitions for the requested transaction in the PG DD.

PGU-35017 missing environment *ename compno* for *t# tin d# din*

Cause: The combined environment, *ename compno*, was not found in the PG DD for transaction ID# *tin* and data ID# *din*. The environment name stored in the transaction entry and the compiler number stored in a call parameter data entry have no environment rows stored in the PG DD environment tables. This message might be preceded by specific messages describing the error.

Action: Ensure that the Oracle server that supports the PG DD is active and that the TRANSACTION ENVIRONMENT name and the DATA LANGUAGE name are defined in the PG DD with supported values. If the error persists, then regenerate the error with tracing enabled. Supported customers can contact Oracle Support Services for assistance.

PGU-35018 failed select for statement *statement, dfmt tfmt:rc rc*

Cause: The information was not fetched from the Oracle dual for SQL statement, *statement*, with date format *dfmt* and time format *tfmt*. The Oracle error code is *rc*. Either the date and time formats that are specified are invalid, or some other Oracle server error occurred. This message is preceded by an Oracle server message for the specific error encountered.

Action: Ensure that the Oracle server that supports PGAU is active. If the error persists, then regenerate the error with tracing enabled. Supported customers can contact Oracle Support Services for assistance.

PGU-39100 EP

Cause: Subroutine call tracing was active. The indicated subroutine was called.

Action: None.

PGU-39101 RP *rc rc*

Cause: Subroutine call tracing active. The indicated subroutine is returned with the shown return code.

Action: None.

PGU-39102 EP *tracef, pkgexf, addr, tname, ver, tipname, dblink, outspec, outbody, pgddver, pgauver*

Cause: Subroutine call tracing active. The GPG main routine was called from PGAU.

Action: None.

PGU-39103 RP *rc rc, rstat exit*

Cause: Subroutine call tracing active. The indicated subroutine is returned from its indicated exit with the shown return code.

Action: None.

PGU-39107 EP *addr*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39108 EP *name, num*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39110 EP *idtype idnum*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39111 EP *f##, dup, varname, vartype, file*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39112 EP *seg=>file*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39113 EP *addr for len bytes*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39114 EP *idtype idnum, idtype idnum, string*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39116 EP *p-addr1,c-addr2*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39117 EP *p-addr1,c-addr2,LDldind*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39118 EP *segpurp, segname, len*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39119 EP *s/r,c#c#,d#d#,f#f#,file*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39120 EP *c-addr1,p-addr2,faddr3*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39121 EP *d#d#,f#f#*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39122 EP *addr, name,id#*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39123 EP *int, int, ptr*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39124 EP *int, int*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39125 EP *int, int, int*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39126 EP *str,int, str, int*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39127 EP *str, int*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39128 EP *ddref, idtype idnum, file*

Cause: Subroutine call tracing active. The indicated subroutine was called.

Action: None.

PGU-39200 Initialization rc *rc*

Cause: Initialization or termination tracing active. Successful initialization was indicated by the subroutine.

Action: None.

PGU-39201 Initialization failure rc *rc*

Cause: Initialization or termination tracing active. Failed initialization was indicated by the subroutine.

Action: None.

PGU-39210 Termination rc rc

Cause: Initialization or termination tracing active. Successful termination was indicated by the subroutine.

Action: None.

PGU-39211 Termination failure rc rc

Cause: Initialization or termination tracing active. Failed termination was indicated by the subroutine.

Action: None.

PGU-39300 QM rc rc, addr

Cause: Queue management tracing active. Successful alteration of the TDT.

Action: None.

PGU-39301 QM failure rc rc, addr

Cause: Queue management tracing active. Failed alteration of the TDT.

Action: None.

PGU-39302 QM p-idtype idnum,c-f###,LDldind

Cause: Queue management tracing active. Successful alteration of the TDT.

Action: None.

PGU-39303 QM warning rc rc, addr

Cause: Queue management tracing active. Requested structure not located.

Action: None.

PGU-39400 IT read len bytes from file

Cause: Initiation or termination tracing active. File or segment data was successfully read.

Action: None.

PGU-39404 IO file file ctlop for mode mode

Cause: I/O tracing active. Successful file control operation.

Action: None.

PGU-39405 IO len ioop bytes file

Cause: I/O tracing active. Successful read or write of file or segment data.

Action: None.

PGU-39500 DD rc rc,addr idtype idnum

Cause: Data Dictionary tracing active. Successful select and load of a PG DD entry.

Action: None.

PGU-39501 DD rc rc - warning

Cause: Data Dictionary tracing active. Select or load of a PG DD entry failed.

Action: None.

PGU-39502 DD rc rc,addr cbtype cmode mask mopt

Cause: Data Dictionary tracing active. Successful select and load of a format conversion block.

Action: None.

PGU-39503 OC *statement curop curno*

Cause: Oracle call tracing active. Successful cursor operation.

Action: None.

PGU-39504 DD *rc rc, uid, date, time*

Cause: Data Dictionary tracing active. Successful select from dual for current session attributes.

Action: None.

PGU-39510 OC *curop hvar cursor curno statement, rc upirc*

Cause: Oracle call tracing active. Successful BIND variable operation.

Action: None.

PGU-39512 OC *curop hvar cursor curno statement, rc upirc*

Cause: Oracle call tracing active. Successful DEFINE operation.

Action: None.

PGU-39513 OC *curop cursor curno statement, rc upirc*

Cause: Oracle call tracing active. Successful EXECUTE operation.

Action: None.

PGU-39514 OC *curop cursor curno statement, rows rows,rc upirc*

Cause: Oracle call tracing active. Successful FETCH operation.

Action: None.

PGU-39600 TG TIP *tipname,SFD tipcall*

Cause: TIP package generation tracing active. Successful generation of TIP specification function definition.

Action: None.

PGU-39601 TG TIP *tipname,SVD tipvar*

Cause: TIP package generation tracing active. Successful generation of TIP specification variable definition.

Action: None.

PGU-39602 TG *oper for type=idno qual1,qual2 l=len o=offset*

Cause: TIP package generation tracing active. Successful generation of TIP data conversion logic. This trace entry is written at the completion of the generation for every data parameter of a TIP call and for every field of a data parameter. There should be one trace for every parameter exchanged by the TIP and one trace for every data field of each parameter.

Action: None.

PGU-39603 TG DBlink *gw,rpc rpc*

Cause: TIP package generation tracing active. Successful generation of specified PGA gateway server RPC.

Action: None.

PGU-39604 TG *edit var var PLS pls*

Cause: TIP package generation tracing active. Successful generation of specified PL/SQL variable conversion.

Action: None.

PGU-39605 TG F-chk name type f## fl=flen typepid# pl=plen

Cause: TIP package generation tracing active. Successful syntax check of COBOL field.

Action: None.

PGU-39606 TG type => item

Cause: TIP package generation tracing active. Successful syntax check of COBOL field. This trace entry is supplemental to message PGU-39605.

Action: None.

PGU-39607 TG type => a#a# name value

Cause: TIP package generation tracing active. Successful syntax check of COBOL field numeric attribute. This trace entry is supplemental to message PGU-39605.

Action: None.

PGU-39608 TG type => a#a# name value

Cause: TIP package generation tracing active. Successful syntax check of COBOL field numeric attribute. This trace entry is supplemental to message PGU-39605.

Action: None.

PGU-39609 TG type => a#a# name value

Cause: TIP package generation tracing active. Successful syntax check of COBOL field date attribute. This trace entry is supplemental to message PGU-39605.

Action: None.

PGU-39610 TG oper for type#idno qual1, type

Cause: TIP package generation tracing active. Successful generation of TIP function. This trace entry is written at the completion of generation for every TIP public function. There should be one trace of this type for every CALL defined in the TRANSACTION.

Action: None.

PGU-39620 TG rc rc, num

Cause: TIP package generation tracing active. Successful calculation of alignment or offset computation.

Action: None.

Messages PGU-41000 to PGU-41119

PGU-41000 Invalid field definition syntax

Cause: A field definition was invalid. There might be invalid punctuation or an invalid keyword present.

Action: Delete invalid punctuation. Replace any invalid keywords.

PGU-41001 unexpected end of input

Cause: An unexpected end of file condition occurred in the PGAU command stream.

Action: Check the `PGAU` statement for proper syntax and premature ending on the last line.

PGU-41002 improper REPORT statement

Cause: A syntax error in occurred the `REPORT` statement.

Action: Check for misspelled items or unbalanced parentheses within the `REPORT` statement.

PGU-41003 invalid data field definition name

Cause: A `DEFINE` or `REDEFINE DATA` statement has specified an invalid name (`field`) or invalid punctuation for the field definition. A field name must begin with an alphabetic character and contain only alphanumeric characters or an underscore (`_`). Only syntax checking continues with the next statement.

Action: Correct the specification of the field name or its delimiters. Supply the field name if it is missing.

PGU-41004 invalid delimiter

Cause: An invalid delimiter was found, such as a parenthesis or comma in the wrong place.

Action: Delete the delimiter it required, or check if there are missing tokens before a delimiter.

PGU-41005 invalid PGAU statement syntax

Cause: A token was found that is not a valid `PGAU` keyword, a delimiter, or an identifier name. Only syntax checking continues with the next statement.

Action: Correct the `PGAU` statement.

PGU-41006 value for parameter 'keyword' missing

Cause: The parameter keyword (`keyword`) was correctly specified, but the parameter value was not found. Only syntax checking continues with the next statement.

Action: Supply the missing parameter value, or remove the keyword from the `PGAU` statement. `PGAU` parameter values should be enclosed in parentheses immediately following the keyword with no intervening space.

PGU-41007 parameter 'dupparm' was previously specified

Cause: The `dupparm` parameter was specified earlier in a `PGAU` statement and is only allowed once. Only syntax checking continues with the next statement.

Action: Remove one of the duplicated specifications from the statement.

PGU-41008 invalid DATA definition dname specified

Cause: An invalid token has been specified for the data name for a `DEFINE DATA` or `REDEFINE DATA` statement. A `DATA` identifier (`dname`) must begin with an alphabetic character and must contain only alphanumeric characters. Only syntax checking continues with the next statement.

Action: Correct the `DATA` identifier `dname` in the statement.

PGU-41012 invalid PLSDNAME parameter

Cause: The parameter specified in the `PLSDNAME` clause of a `DEFINE` or `REDEFINE DATA` statement does not comply with `PL/SQL` syntax for `PL/SQL` variable names. Only syntax checking continues with the next statement.

Action: Correct the `PLSDNAME` parameter to use a name that complies with `PL/SQL` syntax for data variables.

PGU-41013 invalid LANGUAGE parameter

Cause: The LANGUAGE parameter in a DEFINE or REDEFINE DATA statement is not a valid PGA compiler language. Only syntax checking continues with the next statement.

Action: Correct the LANGUAGE parameter to specify a valid PGA compiler language. The only valid language is IBMVSCOBOLII.

PGU-41014 invalid USAGE parameter

Cause: The USAGE parameter in a DEFINE or REDEFINE DATA statement is not a valid PGAU usage. Only syntax checking continues with the next statement.

Action: Correct the USAGE parameter to specify a valid use. Valid use choices are SKIP, ASIS, PASS and NULL.

PGU-41015 USAGE parameter was previously specified

Cause: The USAGE parameter was specified more than once in a DEFINE or REDEFINE DATA statement. This parameter can be specified only once for each data definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of USAGE from the PGAU statement.

PGU-41016 LANGUAGE parameter was previously specified

Cause: The LANGUAGE parameter was specified more than once in a DEFINE or REDEFINE DATA statement. This parameter can be specified only once for each data definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of LANGUAGE from the PGAU statement.

PGU-41017 PLSDDNAME parameter was previously specified

Cause: The PLSDDNAME parameter was specified in a DEFINE or REDEFINE DATA statement. This parameter can be specified only once for each data definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of PLSDDNAME from the statement.

PGU-41018 invalid COMPOPTS parameter

Cause: The COMPOPTS parameter in a DEFINE or REDEFINE DATA statement is not a valid compiler option string recognized by PGAU. Only syntax checking continues with the next statement.

Action: Correct the COMPOPTS parameter to specify valid options. The only valid option is TRUNC (BIN).

PGU-41019 COMPOPTS parameter was previously specified

Cause: The COMPOPTS parameter was specified more than once in a DEFINE or REDEFINE DATA statement. This parameter can be specified only once, for each data definition. Syntax checking only continues with the next statement.

Action: Remove the second instance of COMPOPTS from the PGAU statement.

PGU-41020 invalid CALL definition cname specified

Cause: The token was not a valid call identifier name for a DEFINE CALL statement. A CALL identifier (cname) must begin with an alphabetic character and contain only alphanumeric characters. Only syntax checking continues with the next statement.

Action: Correct the CALL identifier (cname) in the statement.

PGU-41021 PKGCALL parameter was previously specified

Cause: The `PKGCALL` parameter `keyword`, was specified more than once in a `DEFINE CALL` statement. This parameter can be specified only once for each call definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of `PKGCALL` from the statement.

PGU-41022 invalid PKGCALL parameter

Cause: The function name specified in the `PKGCALL` parameter of a `DEFINE CALL` statement does not comply with PL/SQL syntax for PL/SQL function names. Only syntax checking continues with the next statement.

Action: Correct the `PKGCALL` parameter to use a name that complies with PL/SQL syntax for function calls.

PGU-41025 invalid VERSION number specified

Cause: The token was not a valid version number in a `PGAU DEFINE, UNDEFINE CALL, TRANSACTION, or GENERATE` statement. Valid version numbers are all numeric and must match the Oracle sequence object value reported by a previous `DEFINE` statement. Only syntax checking continues with the next statement.

Action: Correct the `VERSION` number in the statement.

PGU-41026 PARMS parameter was previously specified

Cause: The `PARMS` parameter was specified more than once in a `DEFINE CALL` statement. This parameter can be specified only once for each call definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of `PARMS` from the statement.

PGU-41029 invalid TRANSACTION CALL cname specified

Cause: The token was not a valid call identifier name for specifying Calls in a `DEFINE TRANSACTION` statement. A `CALL` identifier (`cname`) must begin with an alphabetic character and contain only alphanumeric characters and it must have originated in a previous `DEFINE CALL` statement. Only syntax checking continues with the next statement.

Action: Correct the `TRANSACTION CALL` parameter (`cname`) in the statement.

PGU-41030 NLS_LANGUAGE parameter was previously specified

Cause: The `NLS_LANGUAGE` parameter was specified more than once in a `DEFINE TRANSACTION` statement. This parameter can be specified only once for each transaction definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of `NLS_LANGUAGE` from the statement.

PGU-41031 parameter 'parm' specifies undefined Oracle NLS name 'nlsname'

Cause: The `PGAU` parameter (`parm`) specifies an Oracle NLS language name (`nlsname`) that is not defined on the Oracle server to which `PGAU` is connected. The language (`nlsname`) is not defined, possibly because it was not selected when Oracle was installed, or the `GENERATED TIPS` are intended to run on another Oracle.* or it is not a valid Oracle NLS Language, because it is misspelled or unsupported. Only syntax checking continues with the next statement.

Action: Correct the parameter value to specify a valid Oracle NLS Language, or install the language, or ensure the `TIPS` execute on a server which has the NLS language installed, (in which case this message can be considered a warning). Valid languages must begin with an alphabetic character, contain only alphanumeric characters, and not exceed 40 characters in length. Supported customers can contact Oracle Support Services for assistance with installing or determining NLS language support for your platform.

PGU-41032 ENVIRONMENT parameter was previously specified

Cause: The ENVIRONMENT parameter was specified more than once in a DEFINE TRANSACTION statement. This parameter can be specified only once for each transaction definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of ENVIRONMENT from the statement.

PGU-41033 invalid ENVIRONMENT parameter

Cause: The ENVIRONMENT parameter in a DEFINE TRANSACTION statement was not a valid PGA ENVIRONMENT. Only syntax checking continues with the next statement.

Action: Correct the ENVIRONMENT parameter to specify a valid PGA ENVIRONMENT. The only valid environment is IBM370.

PGU-41034 SIDEPROFILE parameter was previously specified

Cause: The SIDEPROFILE parameter was specified more than once in a DEFINE TRANSACTION statement. This parameter can be specified only once for each transaction definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of SIDEPROFILE from the statement.

PGU-41035 invalid SIDEPROFILE name specified

Cause: The token was not a valid name for specifying an SNA Side Profile in a DEFINE TRANSACTION statement. A Side Profile name must begin with an alphabetic character and contain only alphanumeric characters. Only syntax checking continues with the next statement.

Action: Correct the TRANSACTION SIDEPROFILE parameter in the statement

PGU-41036 LUNAME parameter was previously specified

Cause: The LUNAME parameter was specified more than once in a DEFINE TRANSACTION statement. This parameter can be specified only once for each transaction definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of LUNAME from the statement.

PGU-41037 invalid LUNAME name specified

Cause: The token was not a valid name for specifying an SNA LU name in a DEFINE TRANSACTION statement. An LU name must begin with an alphabetic or national (\$, #, @) character and contain only alphanumeric or national characters. If the LU name is fully qualified, then it can contain a single period (.) separating the network name from the LU name. Only syntax checking continues with the next statement.

Action: Correct the TRANSACTION LUNAME parameter in the statement.

PGU-41038 TPNAME parameter was previously specified

Cause: The TPNAME parameter was specified more than once in a DEFINE TRANSACTION statement. This parameter can be specified only once for each transaction definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of TPNAME from the statement.

PGU-41039 invalid TPNAME name specified

Cause: The token was not a valid name for specifying an SNA TP name in a DEFINE TRANSACTION statement. A TP name must begin with an alphabetic or national (\$, #, @) character and contain only alphanumeric or national characters. If the TP name must include a UNIX path specification, then it can contain any

characters valid in a UNIX directory or file name, but it must be enclosed in quotes. Only syntax checking continues with the next statement.

Action: Correct the `TRANSACTION TPNAME` parameter in the statement.

PGU-41040 LOGMODE parameter was previously specified

Cause: The `LOGMODE` parameter was specified more than once in a `DEFINE TRANSACTION` statement. This parameter can be specified only once for each transaction definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of `LOGMODE` from the statement.

PGU-41041 invalid LOGMODE name specified

Cause: The token was not a valid name for specifying an SNA logmode in a `DEFINE TRANSACTION` statement. A log mode must begin with an alphabetic or national (\$, #, @) character and contain only alphanumeric or national characters. Only syntax checking continues with the next statement.

Action: Correct the `TRANSACTION LOGMODE` parameter in the statement.

PGU-41042 SYNCLEVEL parameter was previously specified

Cause: The `SYNCLEVEL` parameter was specified more than once in a `DEFINE TRANSACTION` statement. This parameter can be specified only once for each transaction definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of `SYNCLEVEL` from the statement.

PGU-41043 invalid SYNCLEVEL value specified

Cause: The token was not a valid value for specifying an APPC `SYNCLEVEL` in a `DEFINE TRANSACTION` statement. The only values allowed for `SYNCLEVEL` are 0 and 1. Only syntax checking continues with the next statement.

Action: Correct the `TRANSACTION SYNCLEVEL` parameter in the statement.

PGU-41044 invalid TRANSACTION name

Cause: The token is not a valid transaction identifier name in a `DEFINE TRANSACTION` statement. A transaction name must begin with an alphabetic character and contain only alphanumeric characters. Only syntax checking continues with the next statement.

Action: Correct the `TRANSACTION` name parameter in the statement.

PGU-41045 missing call name in DEFINE TRANSACTION

Cause: `PGAU DEFINE TRANSACTION` syntax requires a call name that was not found. Only syntax checking continues with the next statement.

Action: Correct the `DEFINE TRANSACTION` statement to specify a call name.

PGU-41046 SIDEPROFILE omitted: all of LUNAME, TPNAME, and LOGMODE required

Cause: If the `SNA SIDEPROFILE` parameter was not specified, then the `LUNAME`, `TPNAME`, and `LOGMODE` parameters must be specified in a `DEFINE TRANSACTION` statement. Only syntax checking continues with the next statement.

Action: Correct the `DEFINE TRANSACTION` statement to specify either the `SIDEPROFILE` parameter, the `LUNAME`, `TPNAME` and `LOGMODE` parameters, or the `SIDEPROFILE` parameter and any combination of the `LUNAME`, `TPNAME`, and `LOGMODE` parameters.

PGU-41048 CALL parameter was previously specified

Cause: The `CALL` parameter was specified more than once in a `DEFINE TRANSACTION` statement. This parameter can be specified only once for each transaction definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of `CALL` parameter from the statement.

PGU-41049 `DEFINE TRANSACTION` parameter *parm* previously specified

Cause: The `parm` parameter was specified more than once in a `DEFINE TRANSACTION` statement. This parameter can be specified only once for each transaction definition. Only syntax checking continues with the next statement.

Action: Remove the second instance of the `parm` parameter from the `DEFINE TRANSACTION` statement.

PGU-41053 `VERSION` parameter was previously specified

Cause: The `VERSION` parameter was specified more than once but this parameter can be specified only once in a `GENERATE` or `REDEFINE` statement. Only syntax checking continues with the next statement.

Action: Remove the second instance of `VERSION` from the statement.

PGU-41054 missing version number in `VERSION` parameter

Cause: The version number was missing. The version number must be specified with the `VERSION` keyword enclosed in parenthesis in a `GENERATE` or `REDEFINE` statement. Only syntax checking continues with the next statement.

Action: Correct the version number specification in the `VERSION` parameter.

PGU-41055 invalid `FIELD` or `PLSFNAME` name for indicated language

Cause: The token was not a valid field identifier name for specifying a `FIELD` or `PLSFNAME` parameter in a `REDEFINE DATA` statement. A field identifier name must begin with an alphabetic character and contain only alphanumeric characters. Only syntax checking continues with the next statement.

Action: Correct the `FIELD` or `PLSFNAME` identifier name in the statement.

PGU-41058 invalid transaction name specified

Cause: The token was not a valid transaction identifier name for a `GENERATE` statement. A transaction identifier name must begin with an alphabetic character and contain only alphanumeric characters. In addition, it must originate in a previous `DEFINE TRANSACTION` statement. Only syntax checking continues with the next statement.

Action: Correct the transaction identifier name in the statement.

PGU-41059 `PKGNAME` parameter was previously specified

Cause: The `PKGNAME` parameter was specified more than once in a `GENERATE` statement. This parameter can be specified only once for each `GENERATE` request. Only syntax checking continues with the next statement.

Action: Remove the second instance of `PKGNAME` from the statement.

PGU-41060 invalid `PKGNAME` name specified

Cause: The token was not a valid identifier name for specifying the `PL/SQL TIP` package name in a `GENERATE` statement. A package name identifier must begin with an alphabetic character and contain only alphanumeric characters. Only syntax checking continues with the next statement.

Action: Correct the `GENERATE` `PKGNAME` parameter in the statement.

PGU-41061 `OUTFILE` parameter was previously specified

Cause: The `OUTFILE` parameter was specified more than once in a `GENERATE` statement. This parameter can be specified only once for each `GENERATE` request. Only syntax checking continues with the next statement.

Action: Remove the second instance of `OUTFILE` from the statement.

PGU-41063 DIAGNOSE parameter was previously specified

Cause: The `DIAGNOSE` parameter has already been specified more than once in a `GENERATE` statement and is only allowed once for each `GENERATE` request. Only syntax checking continues with the next statement.

Action: Remove the second instance of `DIAGNOSE` from the statement.

PGU-41064 PGANODE parameter was previously specified

Cause: The `PGANODE` parameter was specified more than once in a `GENERATE` statement. This parameter can be specified only once for each `GENERATE` request. Only syntax checking continues with the next statement.

Action: Remove the second instance of `PGANODE` from the statement.

PGU-41065 invalid PGANODE name specified

Cause: The token was not a valid PGA node identifier name in a `GENERATE` statement. A PGA node identifier name must begin with an alphabetic character and contain only alphanumeric characters, and it must be a valid Oracle database link name. Only syntax checking continues with the next statement.

Action: Correct the `PGANODE` parameter in the statement.

PGU-41068 unable to open the input definition file

Cause: An error occurred while opening the input definition file specified with the `INFILE` parameter in a `DEFINE` or `REDEFINE DATA` statement. Only syntax checking continues with the next statement.

Action: Ensure that the input definition file exists and that read access is allowed to the input definition file.

PGU-41070 COBOL inline definition must start with left-paren at line end

Cause: A COBOL inline definition starts with a left parenthesis. The left parenthesis must be the last character on the line preceding any COBOL data definition statements.

Action: Ensure that an inline COBOL data definition starts on a line following a left parenthesis.

PGU-41071 missing LANGUAGE parameter

Cause: `PGAU DEFINE` or `REDEFINE DATA` syntax requires a `LANGUAGE` parameter that was not found. Only syntax checking continues with the next statement.

Action: Correct the `DEFINE` or `REDEFINE DATA` statement to specify a `LANGUAGE` parameter.

PGU-41072 OPTIONS parameter was previously specified

Cause: The `OPTIONS` parameter was specified more than once in a `GENERATE` statement. This parameter can be specified only once for each `GENERATE` request. Only syntax checking continues with the next statement.

Action: Remove the second instance of `OPTIONS` from the statement.

PGU-41073 repeated OPTIONS subparameter

Cause: The indicated subparameter was previously specified in an `OPTIONS` parameter. The only valid `OPTIONS` subparameter is `UDF`. It can be specified only once for each `GENERATE` statement. Only syntax checking continues with the next statement.

Action: Remove the duplicate subparameter from the `OPTIONS` specification.

PGU-41075 `TRACE` parameter was previously specified

Cause: The `TRACE` parameter was specified more than once in a `GENERATE` statement. This parameter can be specified only once for each `GENERATE` request. Only syntax checking continues with the next statement.

Action: Remove the second instance of `TRACE` from the statement.

PGU-41076 repeated `TRACE` or `PKGEX` parameter

Cause: The indicated parameter was previously specified in a `DIAGNOSE TRACE` or `PKGEX` parameter. The valid parameters are `SE`, `IT`, `QM`, `IO`, `DD`, `TG`, `OC`, `DC` and `DR`, and each can be specified only once for each `GENERATE` statement. Only syntax checking continues with the next statement.

Action: Remove the duplicate parameter from the `TRACE` or `PKGEX` specification.

PGU-41077 `PKGEX` parameter was previously specified

Cause: The `PKGEX` parameter was specified more than once in a `GENERATE` statement. This parameter can be specified only once for each `GENERATE` request. Only syntax checking continues with the next statement.

Action: Remove the second instance of `PKGEX` from the statement.

PGU-41079 data definition *dname* version *dver* inserted into PG DD

Cause: PGAU inserted the data definition, *dname*, with the version number, *dver*, as the result of processing a `DEFINE DATA` statement.

Action: Note the `VERSION` number of the data definition for future reference.

PGU-41080 call definition *cname* version *cver* inserted into PG DD

Cause: PGAU inserted the call definition, *cname*, with the version number, *cver*, as the result of processing a `DEFINE CALL` statement.

Action: Note the `VERSION` number of the call definition for future reference.

PGU-41081 transaction definition *tname* version *tver* inserted into PG DD

Cause: PGAU inserted the transaction definition, *tname*, with the version number, *tver*, as the result of processing a `DEFINE TRANSACTION` statement.

Action: Note the `VERSION` number of the transaction definition for future reference.

PGU-41082 data definition *dname* version *dver* updated in PG DD

Cause: PGAU updated the data definition, *dname*, with the version number, *dver*, as the result of processing a `REDEFINE DATA` statement.

Action: None required. This is an informational message.

PGU-41083 data definition *dname* version *dver* deleted from PG DD

Cause: PGAU deleted the data definition, *dname*, with the version number, *dver*, as the result of processing an `UNDEFINE DATA` statement.

Action: None required. This is an informational message.

PGU-41084 call definition *cname* version *cver* deleted from PG DD

Cause: PGAU deleted the call definition, *cname*, with the version number, *cver*, as the result of processing an UNDEFINE CALL statement.

Action: None required. This is an informational message.

PGU-41085 transaction definition *tname* version *tver* deleted from PG DD

Cause: PGAU deleted the transaction definition, *tname*, with the version number, *tver*, as the result of processing an UNDEFINE TRANSACTION statement.

Action: None required. This is an informational message.

PGU-41086 length of *var* token exceeds maximum allowed length of *len*

Cause: The object name of type *var* was longer than the maximum length *len* allowed by PGAU. Only syntax checking continues with the next statement.

Action: Reduce the length of the indicated name.

PGU-41087 data name *dname* does not exist

Cause: A DEFINE CALL referenced a data definition name (*dname*) that does not exist.

Action: Use only previously defined data definitions in a DEFINE CALL statement.

PGU-41088 data name *dname* or specified version (*dver*) of data name *dname* does not exist

Cause: A DEFINE CALL referenced a data definition name (*dname*) that does not exist, or specified a nonexistent version (*dver*) of the data definition (*dname*).

Action: Use only previously defined data definitions in a DEFINE CALL statement.

PGU-41089 call name *cname* does not exist

Cause: A DEFINE TRANSACTION referenced a call definition name (*cname*) that does not exist.

Action: Use only previously defined call definitions in a DEFINE TRANSACTION statement.

PGU-41090 call name *cname* or specified version (*cver*) of call name *cname* does not exist

Cause: A DEFINE TRANSACTION referenced a call definition name (*cname*) that does not exist, or specified a nonexistent version (*cver*) of the call definition (*cname*).

Action: Use only previously defined call definitions in a DEFINE TRANSACTION statement.

PGU-41091 data name *dname* does not exist

Cause: An UNDEFINE DATA referenced a data definition name (*dname*) that does not exist.

Action: Use only previously defined data definitions in an UNDEFINE DATA statement.

PGU-41092 data name *dname* or specified version (*dver*) of data name *dname* does not exist

Cause: An UNDEFINE DATA referenced a data definition name (*dname*) that does not exist, or specified a non-existent version (*dver*) of the data definition (*dname*).

Action: Use only previously defined data definitions in an `UNDEFINE DATA` statement.

PGU-41093 call name *cname* does not exist

Cause: An `UNDEFINE CALL` referenced a call definition name (*cname*) that does not exist.

Action: Use only previously defined call definitions in an `UNDEFINE CALL` statement.

PGU-41094 call name *cname* or specified version (*cver*) of call name *cname* does not exist

Cause: An `UNDEFINE CALL` referenced a call definition name (*cname*) that does not exist, or specified a nonexistent version (*cver*) of the call definition (*cname*).

Action: Use only previously defined call definitions in an `UNDEFINE CALL` statement.

PGU-41095 transaction name *tname* does not exist

Cause: An `UNDEFINE TRANSACTION` referenced a transaction definition name (*tname*) that does not exist.

Action: Use only previously defined transaction definitions in an `UNDEFINE TRANSACTION` statement.

PGU-41096 transaction name *tname* or specified version (*tver*) of transaction name *tname* does not exist

Cause: An `UNDEFINE TRANSACTION` referenced a transaction definition name (*tname*) that does not exist, or specified a non-existent version (*tver*) of the transaction definition (*tname*).

Action: Use only previously defined transaction definitions in an `UNDEFINE TRANSACTION` statement.

PGU-41097 language parameter given in `REDEFINE DATA` conflicts with `DEFINE DATA`

Cause: The `LANGUAGE` parameter in a `REDEFINE DATA` statement for a given data name specified a different language from the one originally specified in the `DEFINE DATA` statement.

Action: Change the `LANGUAGE` parameter in the `REDEFINE DATA` statement to that of the original `DEFINE DATA` statement. The associated language-dependent field definitions must also be changed.

PGU-41100 `FIELD` parameter on `REDEFINE` specifies unknown field name (*fname*)

Cause: A `REDEFINE` with the `FIELD` parameter specified a field name, (*fname*), that does not currently exist in the data object being redefined.

Action: Specify the correct field name on the `FIELD` parameter. Use the `REPORT DATA` statement to list current field names.

PGU-41101 `INFILE` parameter given on `DEFINE DATA` with inline data

Cause: An `INFILE` parameter was found on a `DEFINE DATA` statement, but an inline data definition was also found. Only one of the two forms is allowed in a `DEFINE DATA` statement.

Action: Remove the `INFILE` instance or the inline data definition.

PGU-41102 data definition missing on *inline DATA*

Cause: Neither an inline data definition nor an `INFILE` parameter was found in a `DEFINE DATA` or `REDEFINE DATA` statement. One of these is required in a `DEFINE DATA` statement. One of these might be required in a `REDEFINE DATA` statement, depending on which other parameters are specified.

Action: Specify either an `INFILE` parameter or an inline data definition.

PGU-41103 INFILE parameter previously specified

Cause: The `INFILE` parameter was specified more than once in a `DEFINE DATA` or `REDEFINE DATA` statement. This parameter can be specified only once for each data definition or redefinition. Only syntax checking continues with the next statement.

Action: Remove the second instance of `INFILE` from the statement.

PGU-41105 memory exhausted

Cause: Memory could not be allocated.

Action: Supported customers can contact Oracle Support Services for assistance.

PGU-41106 missing left parenthesis in a GROUP

Cause: A left parenthesis that must immediately follow a `GROUP` verb was missing.

Action: Examine the source for the missing left parenthesis.

PGU-41107 GROUP within a GROUP not allowed

Cause: A `GROUP` was found within a `GROUP`.

Action: Delete the inner `GROUP` along with its enclosing parentheses.

PGU-41108 data definition *dataname* does not exist

Cause: A data name was specified but does not exist.

Action: Check the data name for a misspelling. Or if an explicit version was specified, then recheck for misnumbering. Use the `REPORT` verb to find out what exists.

PGU-41109 field name *fieldname* does not exist in data definition *dataname*

Cause: A `REDEFINE` statement for a field specified a field that does not exist in the given data definition.

Action: Check the data name or field name for misspelling. Or if an explicit version was specified, then recheck for misnumbering. Use the `REPORT` verb to find out what exists.

PGU-41110 field name *fieldname* appears multiple times in data definition *dataname*

Cause: A `REDEFINE` specified a field name that was defined more than once.

Action: Replace the ambiguous field name with a qualified field name. An example of a qualified field name is `FIELD1.SUBFIELD2`. It might be easier to just `REPORT` the `DATA` designation into an output file, edit the output file and `REDEFINE` the `DATA` object.

PGU-41111 missing *token* name

Cause: A `DEFINE` or `REDEFINE` statement did not include the name of a transaction object or a call object. The type of the object is *token*.

Action: Supply the missing name as well as any other parameters that are needed.

PGU-41112 invalid name for UNDEFINE *token*

Cause: The identifier name in an UNDEFINE statement was invalid. It must begin with an alphabetic character and contain only alphanumeric characters. The type of the object is *token*.

Action: Check the identifier name for invalid characters and correct them.

PGU-41113 invalid WITH operand

Cause: The token was not a valid operand in a WITH phrase. Only DATA and CALL statements are valid operands.

Action: Ensure that a WITH phrase is followed by a DATA or CALL operand.

PGU-41114 duplicate WITH operand

Cause: An identical WITH phrase was found in this statement.

Action: Delete one of the identical phrases.

PGU-41115 invalid WITH operand in this context

Cause: A WITH phrase was found in an invalid context. The phrase, WITH CALLS, can appear only in an UNDEFINE TRANSACTION statement. The phrase, WITH DATA, cannot appear in an UNDEFINE DATA statement.

Action: Delete the invalid WITH phrase.

PGU-41119 expecting one of DATA, CALL, or TRANSACTION in a DEFINE

Cause: A DATA, CALL, or TRANSACTION was missing and the type of object being defined could not be specified.

Action: Insert a DATA, CALL, or TRANSACTION operand in the command.

Messages PGU-41120 to PGU-42042

PGU-41120 VERSION parameter conflicts with UNDEFINE ALL

Cause: A VERSION parameter was found in an UNDEFINE ALL.

Action: Either delete the ALL (in UNDEFINE ALL), or delete the VERSION parameter.

PGU-41121 invalid DIAGNOSE parameter

Cause: An invalid subparameter was found in a DIAGNOSE parameter.

Action: Check the subparameter for misspelling.

PGU-41122 no transaction name given in a GENERATE statement

Cause: The transaction name was missing.

Action: Specify a transaction name.

PGU-41123 expecting DATA (type of object)

Cause: At this point in the REDEFINE, token was expected.

Action: Insert *token* into the REDEFINE statement.

PGU-41124 expecting one of DATA, CALL, or TRANSACTION in an UNDEFINE

Cause: Either DATA, CALL, or TRANSACTION must be specified to identify the type of object to be deleted.

Action: Insert either DATA, CALL, or TRANSACTION into the command.

PGU-41125 expecting the name of a *token* object

Cause: The type of object (DATA, CALL, or TRANSACTION) must be followed by a name for the object, which was coming.

Action: Insert the name of a DATA, CALL, or TRANSACTION object.

PGU-41126 WITH token immediately followed by another WITH token

Cause: There were two successive WITH tokens in the command input stream.

Action: Insert DATA or CALLS after the WITH token, depending on the type of referenced objects must be undefined or reported.

PGU-41127 a DATA or CALLS token was found but was not preceded by a WITH token

Cause: There is a missing WITH token before DATA or CALLS.

Action: Insert the WITH token in the appropriate place.

PGU-41128 a WITH token ends *statement*

Cause: A WITH token was found as the last token in an UNDEFINE or REPORT command.

Action: Insert DATA or CALLS after the WITH token to delete or report all referenced DATA or CALLS.

PGU-41129 parameter *parm* invalid without FIELD parameter

Cause: The `parm` parameter was specified without the FIELD parameter. The FIELD parameter, is a prerequisite to specify the `parm` parameter because `parm` applies to FIELDS within a DATA redefinition. PGAU REDEFINE parameters that require FIELD are:

PLSFNAME
CODEPAGE

Action: Include the FIELD parameter in the REDEFINE statement to indicate to the FIELD to which the redefinition of `parm` applies, or remove the `parm` specification.

PGU-41131 FIELD specified, but USAGE, PLSFNAME, or language input absent

Cause: The keyword, which the FIELD parameter must be accompanied by , was missing. These parameters are as follows:

USAGE
PLSFNAME
INFILE

Action: Ensure that *at least one* of the keywords specified in this example is specified.

PGU-41132 language input defines more than one field

Cause: A REDEFINE DATA with the FIELD option indicated that a single field was being redefined. However, the language input contained definitions for more than one field.

Action: Delete the extra field definitions.

PGU-41133 language input defines no data field for REDEFINE ... FIELD(...)

Cause: A REDEFINE DATA with the FIELD option indicated that a single field was being redefined. However, the language input contained no field definitions.

Action: Supply the (single) field definition in the language input.

PGU-41134 no *type* PGAU objects satisfied the REPORT request

Cause: A REPORT command requested information about a PGAU data object with a *type* specified by *type*, but an object of that *type* was not found.

Action: This is an informational message. No action is required.

PGU-41135 no PGAU *type* objects named *name* satisfied the REPORT request

Cause: A REPORT command requested information about a PGAU data object with a *type* specified by *type* and named *name*. However, an object with that *name* was not found.

Action: This is an informational message. No action is required.

PGU-41136 no *type* PGAU object named *name* at version *ver* satisfied the REPORT request

Cause: A REPORT command requested information about a specified by PGAU data object with a *type* specified by *type*, a *name* specified by *name*, and a *version* of *ver*. However, an object with that *name* and *version* was not found.

Action: This is an informational message. No action is required.

PGU-41137 too many version requests; maximum of *max* allowed

Cause: The VERSION parameter of a REPORT request had too many subparameters, and *max* is the maximum allowed.

Action: Split the request into multiple REPORT statements.

PGU-41138 missing *object name*

Cause: The *object name* was missing in a REPORT statement.

Action: Supply the missing *object name*.

PGU-41139 use of VERSION parameter conflicts with ALL parameter

Cause: A VERSION parameter was found. But an ALL parameter was also found (specifying all versions).

Action: Delete the ALL parameter or the VERSION parameter.

PGU-41140 version number expected

Cause: A version number was expected at this point.

Action: Supply a version number or delete the entire VERSION parameter.

PGU-41141 an invalid version number was found

Cause: A nonnumeric string or an invalid version number, was found when scanning for a version number.

Action: Check the statement for missing delimiters.

PGU-41142 version parameter has already been specified

Cause: The VERSION parameter already exists and another was specified.

Action: Delete a VERSION parameter so that only one is left.

PGU-41143 this *type* of WITH operand invalid with a *type* object

Cause: The WITH phrase which is invalid with a *type* object, was used. For example, REPORT DATA . . . WITH DATA is invalid.

Action: Delete the invalid WITH phrase.

PGU-41144 default PKGCALL parameter truncated to *value*

Cause: The PKGCALL parameter was defaulted in a DEFINE CALL statement. By default, it takes the value of the CALL object name. However, the length of the CALL object name was greater than the maximum length allowed for the PKGCALL parameter, which is 30. The default value used was *value*.

Action: This is an informational message. No action is required.

PGU-41145 default PLSDNAME parameter truncated to *value*

Cause: The PLSDNAME parameter was defaulted in a DEFINE DATA statement. By default, it takes the value of the DATA object name. However, the length of the DATA object name was greater than the maximum length allowed for the PLSDNAME parameter, which is 30. The default value used was *value*.

Action: This is an informational message. No action is required.

PGU-41146 invalid INFILE parameter

Cause: The INFILE parameter in a DEFINE or REDEFINE DATA statement did not designate a valid filename. Only syntax checking continues with the next statement.

Action: Correct the INFILE parameter to specify a valid file name.

PGU-42001 ALL not valid in VALUE(S) clause at level 88

Cause: ALL specified in a level 88 (conditionnames) entry, and this is invalid.

Action: Delete the reserved word ALL.

PGU-42002 COPY statement ended prematurely

Cause: There were missing operands in the COPY clause.

Action: Insert missing operands.

PGU-42003 COPY reserved word found within a COPY clause

Cause: COPY clause was found embedded within another COPY clause.

Action: Check COPY clauses. The statement terminator might be missing.

PGU-42004 COPY not allowed as operand-1 or operand-2 in COPY REPLACING

Cause: The COPY reserved word was used as operand-1 or operand-2 in the REPLACING clause of a COPY statement, and this is not allowed.

Action: If it is operand-1, then replace COPY with some other word here and in the copied source file. If it is operand-2, then check the offending COPY clause for missing terminator or reserved words.

PGU-42005 COPY statement: nothing to copy

Cause: File name operand was not given in the COPY clause.

Action: Insert filename operand in the COPY clause.

PGU-42006 invalid token *token* in line as follows

Cause: An unrecognized token was found when checking for a COBOL reserved word.

Action: Check source around unrecognized token for other missing reserved word or statement terminator.

PGU-42007 invalid EJECT statement

Cause: An EJECT statement was used with operands.

Action: Delete the extraneous text after EJECT, up to the statement terminator.

PGU-42008 invalid SKIP n statement

Cause: A SKIP1, SKIP2, or SKIP3 statement cannot have operands.

Action: Delete the extraneous text after the SKIP n , up to the statement terminator.

PGU-42009 text of source line

Cause: This is an echo of the offending source line.

Action: Check this line with respect to a previous error message.

PGU-42010 level-number with num digits exceeds maximum of two

Cause: Level numbers must be restricted to a maximum of two digits.

Action: Ensure level number is in range 01 - 49, 66, 77, or 88.

PGU-42011 invalid level number of num found

Cause: Level numbers must be restricted to 01 - 49, 66, 77, or 88.

Action: Ensure level number is in range 01 - 49, 66, 77, or 88.

PGU-42012 level 01 or 77 has yet to be found

Cause: Record definition must start at level 01 or 77. A level other than 01 or 77 was found.

Action: Examine source for missing level 01 or 77 definition. If the offending definition is the top level for a record, then renumber it to level 01 or 77.

PGU-42013 level 01 or 77 not yet been found or begun between columns a and b

Cause: There was a missing level 01 or 77 in the source file, or the 01 or 77 did not start within the columns limit specified by margins A and B.

Action: If level 01 or 77 appears, then ensure that it starts before margin B. If level 01 or 77 does not appear, then a renumbering of levels might be appropriate.

PGU-42014 RENAMES clause must be at level 66

Cause: The RENAMES reserved word must be used only at level 66.

Action: Renumber the level to 66.

PGU-42015 multiple *type* clauses found

Cause: Multiple instances of a clause were found, and multiple instances of the type of clause given are not allowed with a data definition.

Action: Ensure that there is only one instance of the type of clause given.

PGU-42016 invalid figurative_type value passed

Cause: This was an internal error.

Action: Supported customers should contact Oracle Support Services for assistance.

PGU-42017 VALUES can be used only at level 88

Cause: The VALUES keyword can only be specified in a data definition describing a condition and only at level 88.

Action: If the definition describes a condition, then ensure that it is at level 88. If the definition does not describe a condition, then use VALUE IS rather than VALUES ARE.

PGU-42018 could not open input file *filename*

Cause: The specified input file could not be opened for reading.

Action: Ensure that the file exists and allows read access.

PGU-42019 REDEFINES clause should be first clause

Cause: A REDEFINES clause, was not the first clause, and this clause must appear before any other clause in a data definition.

Action: Move the REDEFINES clause to the beginning of the data definition immediately after the data definition name.

PGU-42020 RENAMES *dn-1* (THROUGH) *dn-2* must stand alone

Cause: If a RENAMES clause appears in a data definition, then it must be the ONLY clause.

Action: Delete other clauses.

PGU-42021 unterminated record definition found

Cause: An end of file was found before the ending statement terminator. A common cause for this is a line going beyond column 72 in the COBOL definition.

Action: Check last records in input file for a missing statement terminator.

PGU-42022 VALUES clause must be the only clause at level 88

Cause: The only clause allowed at level 88 was a VALUE or VALUES clause.

Action: Delete any clauses other than a VALUE or VALUES clause at level 88.

PGU-42024 missing period-separator in TITLE statement

Cause: A TITLE statement did not end with a period separator.

Action: Check for missing period separator.

PGU-42025 invalid TITLE statement

Cause: The TITLE statement format was invalid.

Action: Check the format of TITLE statement.

PGU-42026 COPY copies a COPY REPLACING or vice versa

Cause: The source file copied by a COPY statement cannot contain a COPY . . . REPLACING statement. In addition, the source file copied by a COPY . . . REPLACING statement cannot have a COPY statement.

Action: Correct the file in error.

PGU-42027 in line *num* of file *filename* as follows

Cause: This message usually followed another message that describes the actual error. The line following this message echoes the input COBOL source line.

Action: Check messages immediately before and after this message.

PGU-42028 invalid USAGE token: *token*

Cause: An invalid word followed USAGE in a USAGE clause. The word might be valid in COBOL dialects but not in the dialect specified.

Action: Replace operand in USAGE clause with a valid USAGE for the specified COBOL dialect, or specify a different COBOL dialect.

PGU-42029 reserved word *word* is invalid as a PICTURE

Cause: A reserved word was used as a PICTURE operand.

Action: Check PICTURE clause for missing PICTURE operand.

PGU-42030 a *type* clause has been prematurely terminated

Cause: A clause of the type, *type*, has ended, but had missing operands.

Action: Check the clause for missing operands.

PGU-42031 *type* phrase is misplaced after *type* phrase

Cause: A phrase introduced by *type* cannot come after a phrase introduced by *phrase*. Certain phrases must precede other phrases in a data definition. For instance, an ASCENDING [KEY] phrase must appear before an INDEXED clause.

Action: Reverse the placement of the phrases.

PGU-42032 state value *type* is *value*

Cause: An internal error occurred in the PGAU COBOL parser.

Action: Regenerate the error with full diagnostics enabled and save all related input and output files and listings. Supported customers should contact Oracle Support Services for assistance.

PGU-42033 extraneous text at column *col*

Cause: Extraneous text was found at the end of a record definition.

Action: The only source text that can follow the terminating period of an Oracle record definition is a COPY, EJECT, or SKIP*n* statement. Check if the terminating period has been misplaced.

PGU-42034 COPY for file *filename* is recursive

Cause: A COPY statement for the file, *filename*, was found while processing a previous occurrence of a COPY for the same file.

Action: Check the COBOL source for the recursive COPY. A single COPY file may need to be replaced with multiple, uniquely named COPY files to achieve your desired results.

PGU-42035 invalid null operand-1 in pseudo-text in a COPY REPLACING clause

Cause: The first operand in a REPLACING clause was pseudo-text, but the token was null.

Action: Replace the null pseudo-text string with a non-null string.

PGU-42036 invalid input in *type* clause or paragraph

Cause: There was invalid syntax in a DECIMAL-POINT or CURRENCY clause or in an OBJECT-COMPUTER or SOURCE-COMPUTER paragraph.

Action: Check the indicated source lines for syntactical errors.

PGU-42037 missing END-EXEC token

Cause: An END-EXEC was not found while processing an EXEC clause.

Action: Insert an END-EXEC where appropriate. If the COBOL source was generated by another product, then regenerate the source using that product.

PGU-42038 invalid use of reserved word *word*

Cause: A reserved word was found in an illegal context.

Action: Check to see if a reserved word is used where a user data name would normally appear.

PGU-42039 in line as follows

Cause: This message follows another message describing the actual error. The line following this message echoes the input COBOL source line.

Action: Refer to the messages issued immediately before and after this message.

PGU-42040 DEPENDING ON phrase missing in OCCURS clause

Cause: The `DEPENDING ON` clause was missing. An `OCCURS` clause describes a variable length table. The `DEPENDING ON` phrase is required to determine the size of the current instance of the table.

Action: Insert the `DEPENDING ON` clause, or make the table a fixed-length table.

PGU-42041 missing or invalid USAGE clause for DBCS PIC clause

Cause: A `PICTURE` clause specified a `DBCS` datatype, but the required `USAGE IS DISPLAY-1` clause was missing or invalid.

Action: Insert the `USAGE` clause or revise the `PIC` datatype to both required `DBCS`.

PGU-42042 LENGTH IS clause not allowed for numeric or edited data

Cause: A `LENGTH IS` clause was specified for a data item that is not a non edited character data.

Action: Remove the `LENGTH IS` clause or change the `PICTURE` clause for the data item to specify a non edited, character data mask.

pg4tcpmap Tool Messages

This chapter lists the messages issued by the Oracle Procedural Gateway for the TCP/IP mapping tool (`pg4tcpmap`) and provides a possible cause and recommended action for each message.

This chapter contains the following section:

- ["Messages PGU-50001 to PGU-50101"](#) on page 4-1

Messages PGU-50001 to PGU-50101

PGU-50001 Error reading the log file

Cause: An error occurred while reading the log file.

Action: Determine the cause of the problem and correct it before restarting the `pg4tcpmap` tool.

PGU-50002 no data found

Cause: The predicate that was chosen to delete the row from the `PGA_TCP_IMSC` table is not found.

Action: Determine the cause of the problem and correct it before restarting the `pg4tcpmap` tool.

PGU-50003 *parm* length of *len* exceeds the maximum of *max*

Cause: The parameter (`parm`) length (`len`) passed to the `pg4tcpmap` tool was larger than the maximum allowed (`max`).

Action: Correct the parameter in error and re-call the `pg4tcpmap` tool. For more information, refer to the *Oracle Procedural Gateway for APPC User's Guide*.

PGU-50004 `pg4tcpmap` has ended in failure

Cause: An error occurred during processing of the `pg4tcpmap` tool. This message is preceded by additional messages providing more information about the error.

Action: Refer to the messages preceding this one to determine the course of action.

PGU-50101 You have entered an invalid value

Cause: This is a warning that you have entered an invalid value.

Action: The question will be asked again. Enter a valid value.

C

CPI-C

- function
 - gateway error messages, 2-2, 2-3

E

error

- PGA-20910 communication error message, 2-2, 2-3
- PGA-20995 communication error message, 2-10
- PGU-50001, 4-1
- system error number "errno", 2-2
- system error number "errno", 2-3
- error messages, 2-1, 3-1
 - for pg4tcpmap tool, 4-1
 - issued by TIPs, 1-1
 - issued for gateway server, 2-1
 - issued for pg4tcpmap tool, 4-1
 - issued for PGAU, 3-1

G

gateway

- error messages issued for PGAU, 3-1
- error messages issued for the gateway server, 2-1

P

- pg4tcpmap tool messages, 4-1
- PGA_TCP_IMSC table, 4-1
- PGA-20910 message
 - communication error, 2-2, 2-3
- PGA-20995 message
 - communication error, 2-10

S

SNA server

- error messages, 2-2, 2-10
- gateway error PGA-20995, 2-10

T

TIPs

- error messages, 1-1

