



ETI•EXTRACT[®]

*Data System Library Release Notes
for C/ORACLE*

*Release 1.2.6
October, 1997
For ETI•EXTRACT Release 3*

ETI•EXTRACT® Data System Library Release Notes for C/ORACLE

Release 1.2.6, October, 1997

For ETI•EXTRACT 3

THIS DOCUMENT IS THE CONFIDENTIAL AND PROPRIETARY PRODUCT OF EVOLUTIONARY TECHNOLOGIES INTERNATIONAL, INC. ANY UNAUTHORIZED USE, REPRODUCTION, OR TRANSFER OF THIS DOCUMENT IS STRICTLY PROHIBITED. COPYRIGHT © 1997 BY EVOLUTIONARY TECHNOLOGIES INTERNATIONAL, INC. (SUBJECT TO LIMITED DISTRIBUTION AND RESTRICTED DISCLOSURE ONLY.) ALL RIGHTS RESERVED.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Evolutionary Technologies International, Inc.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at 252.227-7013 (48 CFR Ch.2) and/or the Commercial Computer Software Restricted Rights clause at FAR 52.227.19(c). Unpublished: Rights reserved under the copyright laws of the United States. Contractor/manufacturer is:

Evolutionary Technologies International, Inc.
816 Congress Avenue, Suite 1300
Frost Bank Plaza
Austin, Texas 78701

(512) 383-3000

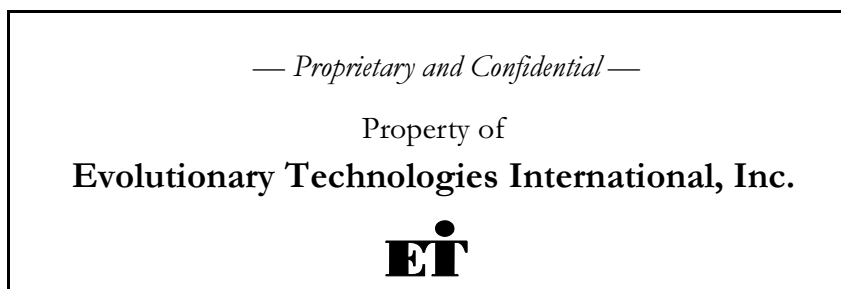
<http://www.eti.com>

ETI•EXTRACT and the ETI logo are registered trademarks and Data System Library is a trademark of Evolutionary Technologies International, Inc.

Adobe and Acrobat are trademarks of Adobe Systems Incorporated.

ORACLE and SQL*Loader are registered trademarks and Pro*C is a trademark of Oracle Corporation, Redwood City, California.

All product names mentioned herein are for identification purposes only and may be trademarks and/or registered trademarks of their respective companies and/or institutions.



Serial Number: **EXT-UNIX-124-00-D-ORA-ENG-RN**

Contents

Introduction	5
Contacting ETI	5
Related Documentation	7
Online Help	7
ETI•EXTRACT Documentation Online	7
Tcl man Pages	8
Note on Retrieve Schema Scripts	8
Problems Resolved	9
Known Problems and Workarounds	12
Known Problems.....	12
Workarounds	12
Host Arrays and Evaluating Results	13
ORACLE Pro*C Pre-Compiler Limitation	16
Changes to Templates in Release 1.2.4.....	17
Changes to DAS Object in Release 1.2.4	18
Other Considerations.....	18

Introduction

These *Release Notes* provide the following information about Release 1.2.6 of the ETI•EXTRACT Data System Library™ (DSL) for C/ORACLE® (for use with ETI•EXTRACT® Release 3):

- A table of problems resolved in Releases 1.2.1, 1.2.2, 1.2.3, and 1.2.4
- Known problems, limitations, and other considerations about this release
- A table of template changes in this release

This release is a maintenance release and primarily contains fixes to reported problems.

Note: This document refers to ETI•EXTRACT Release 3, which includes all releases 3.0 and above.

Contacting ETI

If you encounter difficulty using the DSL for C/ORACLE, have questions regarding the documentation, or have feedback or suggestions that can help ETI improve the DSL for C/ORACLE, please contact the ETI AnswerLine™. ETI places a high value on your success and fulfilling your needs as a customer.

You can reach ETI AnswerLine personnel at:

Customer Area	Phone	Fax	Email
North America	800-856-0416 (toll free) 512-327-6994 ext. 450	512-327-6117	eti.answerline@evtech.com
United Kingdom	01344-382135	01344-382091	eti.answerline@evtech.com
Europe (outside U.K.)	44-1344-382135	44-1344-382091	eti.answerline-europe@evtech.com

ETI's World Wide Web address is <http://www.evtech.com>. You can access the AnswerLink™, ETI's online support center, from there.

Related Documentation

Related documentation for the *Data System Library Release Notes for C/ORACLE* consists of the following:

- *ETI•EXTRACT Data System Library for C/ORACLE Release Notes, Release 1.2.0*
- *ETI•EXTRACT Data System Library for C/ORACLE Release Notes, Release 1.2.1*
- *ETI•EXTRACT Data System Library for C/ORACLE Release Notes, Release 1.2.2*
- *ETI•EXTRACT Data System Library for C/ORACLE Release Notes, Release 1.2.3*
- *ETI•EXTRACT Data System Library Handbook for C/ORACLE, Release 1.2*
- *ETI•EXTRACT Data System Library Installation Guide*
- *ETI•EXTRACT Conversion Specialist's Guide*

- ❑ *ETI•EXTRACT Master User's Guide*
- ❑ *ETI•EXTRACT Reference Manual*
- ❑ *ETI•EXTRACT Installation and Administration Guide*
- ❑ *ETI•EXTRACT Documentation Updates*
- ❑ *The ORACLE documentation library*

Note: A README PDF document may be installed with the other PDF files for the DSL. If so, the README document contains information that became available after the *Data System Library Release Notes for C/ORACLE* was printed.

Online Help

The online help for ETI•EXTRACT is available by using the Help menu in any ETI•EXTRACT tool window, or by pressing the F1 key. You can access general information by clicking **Help** on the Menu Bar. If you want help relative to a task you are performing, press **F1** for context-sensitive help.

ETI•EXTRACT Documentation Online

All of the ETI•EXTRACT documentation is available online in the form of Adobe™ Acrobat Reader™ Portable Document Format (PDF) files. They are located in the doc directory of your ETI•EXTRACT installation directory. The ETI•EXTRACT distribution tape includes a copy of the Unix version of Acrobat Reader, and you can display any of the manuals by using the following command:

```
ex_manual manual_name
```

Substitute the name of a manual for *manual_name*.

Tcl man Pages

You can extend the capabilities of ETI•EXTRACT by writing Program Generator Extension Functions and Grammar Extension Functions in Tool Command Language (Tcl). Tcl is a publicly available interpreted language and is embedded in ETI•EXTRACT. For more information on Tcl, refer to *Practical Programming in Tcl and Tk*, Brent B. Welch, ISBN 0-13-182007-9.

In addition, ETI•EXTRACT includes UNIX manual (man) pages for Tcl. You can view them by entering the command

```
man command_name
```

where *command_name* is a Tcl command.

For example, you can display a summary of Tcl language syntax by entering the command

```
man Tcl
```

Note on Retrieve Schema Scripts

The DSL for C/ORACLE includes three scripts for retrieving schemas. However, `oracle_rs.sql` is the only script that you should run - it automatically determines whether to use

- `oracle6_rs.sql` for ORACLE Version 6 schemas
- `oracle7_rs.sql` for ORACLE Version 7 schemas
- `oracle7_rs.sql` for ORACLE Version 8 schemas

Note: Do *not* run either `oracle6_rs.sql` or `oracle7_rs.sql` directly. Use *only* the `oracle_rs.sql` script. It determines which of the other scripts to invoke.

Problems Resolved

Table 1 describes the reported problems that have been resolved in the DSL for C/ORACLE since Release 1.2. They are listed by Product Design Problem Report (PDPR) number.

Table 1. Problems Resolved in the DSL for C/ORACLE since Release 1.2 (Part 1 of 3)

PDPR	Description
2444*	retrieve-schema process now checks for unsupported datatypes and tables with the same name having different owners; the process notifies the user when either of these situations occurs
2895*	new CVAR named uow_sql_commit_limit added to set the limit for UoW transactions only, thus decoupling them from non-UoW transactions which used the CVAR sql_commit_limit . Three templates are affected: uow-status-struct , uow-commit , and initialize-check-global-cvars .
3103*	an error is now issued if the unit-of-work control file (uow.ctl) defines a unit that does not have mappings defined to it.
3757	filter to set data_conv_mode to data_driven no longer generates spurious messages
3587	Populate programs no longer have misleading code to process NULL characters
3661*	when doing an operation for Modify Value filter, target parts for filter are no longer shown as parts with improper types (for example, a filter for "the product of" or "the sum of" no longer shows parts with a data type of CHAR)
3721	applying SQL filters to dates no longer generates functions in the wrong order
3907	checking for NULL in the target part filter grammar and assigning a value no longer ignores the value
3966	retrieve-schema mechanism has been rewritten from grammar to perl-based with an option for ORACLE Version 6 or 7 support
4037	table lookups where NULL columns are selected no longer cause generation errors
4080*	an error message is not displayed when the Oracle_indexes.txt file is empty (because no data was retrieved)
4150	buffers used for SELECT DATE FROM DUAL code no longer use space padding but do so explicitly as required by ORACLE
4161	subseq no longer returns space when passed NULL but NULL
4240	table lookups no longer generate WHERE ... IS NULL clauses with unresolved CVAR references
4551	table lookup subqueries no longer place all of the right parentheses at the end of compound conditions
4610*	retrieving the sample schema shipped with C/ORACLE 1.2. 4 no longer creates units without parts
4649	setting data_conv_mode to data_driven no longer ignores return codes
4656	table lookups no longer generate faulty code when comparing to dates

Table 1. Problems Resolved in the DSL for C/ORACLE since Release 1.2 (Part 2 of 3)

P DPR	Description
4748	SQL code in Populate programs no longer replaces pound characters (#) in part names with underscore characters (_)
4753	table lookup subqueries no longer fail when applying greater than conditions
4754*	a table lookup where a column is compared to 0 no longer yields an empty result. The template corrected is oracle_populate/ora_proc_table_lookup/lookup-function-where-rhs.
4827	the oracle-fork-strdup-date-format grammar no longer prohibits spaces
4831	target part filter grammar no longer improperly brackets code generated for the Select ... Otherwise path
4846*	sql_error_limit is no longer ignored when the conversion property data_conv_mode is set to data_driven
4849	new lines have been added to runtime messages ETI-ORA-RE-1065 and ETI-ORA-RI-1066 such that they no longer generate text that is hard to capture
4890	target part filter grammar no longer prohibits passing literal arguments to integer functions
5047	target part filter grammar no longer aborts when Spaces for Concatenate is selected and only numeric parts are mapped
5050	target part filter grammar no longer generates faulty code when multi-sourcing a part and choosing a source part to represent the number of spaces to assign to the target part
5125	bulk load .unc script no longer removes special characters in part-name
5203	target unit filter grammar no longer prevents data_conv_mode from being set
5331	value for cip_sql_cursor_end now appears in the proper place
5430	the global variable that holds a table lookup value is no longer re-initialized improperly
5442	Populate programs generated with data_conv_mode set to update_or_insert no longer fail on both the update and the insert (due to a FOR clause being commented out)
5446	the unit-name template that generated a trailing space is no longer used
5523*	the retrieve_schema process no longer generates insufficient FINISH_ARRAY error message
5555	Unit-of-Work Populate programs referencing multiple target tables no longer generate unresolved references in the uow_driver routine
5579	the number of spaces used to fill in filtered target parts is now calculated correctly
5591	the process OracleIndexes.pl program now recognizes both TABLE_NAME and Table
5599*	Modify Value target filter now lists appropriate string parts as valid candidates
5643, 5698, and 3715	dates are no longer improperly loaded when using a target part filter to assign a default value

Table 1. Problems Resolved in the DSL for C/ORACLE since Release 1.2 (Part 3 of 3)

P DPR	Description
5659*	UoW tables with keys no longer change integer fields to chars when mixing standard and scientific notation
5689	a table lookup that tests for NULL no longer gives a generation error
5735	generation of certain warnings no longer causes a Segmentation Fault in populate programs
5750 and 5574	joining two source units in separate databases no longer references an unresolved template, or causes execution of the Query program to loop
5799*	write_allowed flag for each unit in a Populate now gets reset/re-initialized; this affects the process-ifile-for-target-unit-function in oracle_populate.
5866*	the message that is displayed for a referential integrity problem detected during the retrieve-schema process is now more descriptive: "ERROR: There is a Referential Integrity problem in the retrieve schema process. Check Oracle_RI_Dependencies.txt file".
5930	csq1_single_step can now be enabled using ETI*EXTRACT 3.0.2 without generation errors
6102*	the parenthesis error has been corrected in the template oracle_populate/ora_proc_table_lookup/lookup-function-init-hostvar-num-to-str template
6163	the operator in the assignment of unit_status_name.end_of_file has been changed from == to =
6166	the process_oracle_schema.pl program now picks up key designations properly
6314*	virtual parts of type decimal can now be created without receiving an error
6319	the correct version of the retrieve schema script (oracle_rs.sql) is now shipped with the DSL
6392*	retrieve-schema process no longer hangs when it encounters unsupported data types (such as LONG, RAW, or LONGRAW); an error message is displayed instead
6431*	the correct syntax is now generated for oracle_fork_table_lookup_sel operators

* Resolved in Release 1.2.4. All others resolved in Releases 1.2.1, 1.2.2, or 1.2.3.

Known Problems and Workarounds

This section contains information about known problems and workarounds.

Known Problems

The following problems in this release of the DSL for C/ORACLE are known to ETI:

- ❑ Under certain conditions the default Sort intermediate action program (the UNIX **sort** utility) issues messages like the following example and fails to sort the data. ETI is currently working on a resolution to the problem. If you need to use the default Sort program, contact the ETI Answerline for the status of the fix.

Missing NEWLINE at end of file

- ❑ The interaction of ORACLE host array processing with ETI•EXTRACT may result in the following processing problems:
 - **Mistakenly identifying records as causing errors** — If a host array being processed for updates contains an error that causes an ORACLE error (SQLCODE less than zero) and a prior record in the array did not result in a target row being updated (+1403 error), a record will be incorrectly identified as being in error. Additionally, the target row may be updated multiple times with the same values, which would cause incorrect record counts. See the following section titled “Host Arrays and Evaluating Results” for more details.
 - **Incorrect handling of +1403 errors during populate** — When using host arrays for updating or deleting rows, ORACLE returns a single SQLCODE for all the rows processed with a host array. Processing only stops if the SQLCODE is set to a negative (error) value or the array is exhausted. Because of this, +1403 (data not found) warnings cannot be captured on a record basis.

The only circumstance that will actually generate a +1403 error occurs when none of the target rows for the array being processed exist. In this circumstance, the first record in the array is reported as failing when actually all the rows being processed in the array have failed. For more information, see Host Arrays and Result Processing later in this release notes.

- ❑ The retrieve-schema process sometimes relies on relative path names when spawning processes. For example, if the user’s profile has something unexpected in it, such as a CD command, the assumption that the current working directory is msdb/tmp may no longer be valid.
- ❑ The parser can not support recursive execution on phrases.

Workarounds

The following problems have been identified and workarounds have been created to address customer issues:

- Because ETI•EXTRACT can not handle two owners with the same table name, you can run one of the following scripts that check for conditions that the user must edit within the schema prior to retrieval.

```
perl preprocessOracleSchema.txt INFILE OUTFILE
```

where:

```
INFILE is OracleSchema.txt and OUTFILE is tempSchemaError.txt
```

OR

```
processOracleSchema.pl (in the oracle.N directory where N represents the number of the DAS)
```

- When moving from a mainframe database to UNIX or when moving files using ftp, some customers experience problems with end-of-line characters and the trailing /177 at the end of each file. One method for working around this problem is as follows. This method converts /r in the source file (INFILE) to /n in the destination file (OUTFILE). The trailing /177 in the file must be removed manually.

```
tr '\r' '\n' INFILE OUTFILE
```

Host Arrays and Evaluating Results

The DSL for C/ORACLE supports host arrays to improve performance. When using host arrays, ORACLE returns a single SQLCODE for the entire result. Each record in the array is processed until all the records have been exhausted or an SQL error occurs resulting in a negative SQLCODE. After processing the array, the SQLERRD(3) field contains the number of database rows directly affected by the array SQL statement and the SQLCODE contains a +0 if no error occurred, a +1403 if SQLERRD(3) = +0 (or end of cursor on SELECT), or the negative SQLCODE that forced array processing to stop.

Since a +1403 return code means none of the records in the array updated the database, it is impossible to tell from the SQLCODE if an individual record updated the database or not. It is also impossible to tell which record in the host array caused the negative SQLCODE to be returned. The array index is not returned by ORACLE, only the number of rows successfully updated or deleted is returned.

Given these circumstances, when an error occurs the only thing known is the number of records processed successfully. When an error occurs, the Populate program slides the array down by the number of rows successfully processed + 1, reports the last row discarded as an error, and processes the remaining array elements again. For example, assume the array shown in the following table is being processed by an UPDATE statement.

Array Index	Record Number	Scalar SQLCODE	Array SQLCODE	SQLERRD(3)	Records Updated	ORACLE Result	Populate Program Assumes
0	R1	+0				Update	Update
1	R2	+1403				No update	Update
2	R3	+0				Update	Error
3	R4	-1401	-1401	2	2	Error	Not processed
4	R5	+0				Not processed	Not processed
5	R6	+1403				Not processed	Not processed
6	R7	+1403				Not processed	Not processed
7	R8	+0				Not processed	Not processed
8	R9	+0				Not processed	Not processed
9	R10	-1401				Not processed	Not processed

The Scalar SQLCODE column represents the SQLCODE that would have been returned had each row been processed individually. The columns to the right indicate the results after the last row was processed.

ORACLE processing stops with record R4 (array index = 3) because of the -1401 error. The Populate program slides the array down by three (SQLERRD(3) + 1). The last row discarded is assumed to be in error. The new array is processed again, as shown in the following table.

Array Index	Record Number	Scalar SQLCODE	Array SQLCODE	SQLERRD(3)	Records Updated	ORACLE Result	Populate Program Result
0	R4	-1401	-1401	0	2	Error	Error
1	R5	+0				Not processed	Not processed
2	R6	+1403				Not processed	Not processed
3	R7	+1403				Not processed	Not processed
4	R8	+0				Not processed	Not processed
5	R9	+0				Not processed	Not processed
6	R10	-1401				Not processed	Not processed

ORACLE processing stops after the first record and the Populate program creates a new array, as shown in the following table.

Array Index	Record Number	Scalar SQLCODE	Array SQLCODE	SQLERRD(3)	Records Updated	ORACLE Result	Populate Program Result
0	R5	+0				Update	Update
1	R6	+1403				No update	Update
2	R7	+1403				No update	Update
3	R8	+0				Update	Error
4	R9	+0				Update	Not processed
5	R10	-1401	-1401	3	5	Error	Not processed

In this case ORACLE processed all the rows, but since SQLERRD(3) is only 3, the Populate program discards only the first four records and carries forward the last two, as shown in the following table.

Array Index	Record Number	Scalar SQLCODE	Array SQLCODE	SQLERRD(3)	Records Updated	ORACLE Result	Populate Program Result
0	R9	+0				Update applied again	Update
1	R10	-1401	-1401	1	6	Error	Error

In this final array, the update to record R9 is applied again by ORACLE. But since the Populate program is not aware that the record has already been processed, the count of updated records is incremented.

The handling of errors within host arrays may slow down performance due to applying multiple updates to rows, as in the preceding example of record R9. In most situations this should not be the case. However, if a significant number of records are being rejected with negative SQLCODE errors, it may be wise to turn off host array processing by setting the **rows_to_process** property to a value of 1.

C/ORACLE DSL Limitations

This release of the DSL for C/ORACLE has the following limitations:

❑ The following ORACLE data types are not supported:

- LONG
- RAW
- LONGRAW
- MLSLABEL
- ROWID

Note: If you can estimate the maximum size for a LONG data type, there is a workaround described in Chapter 2 of the *ETI•EXTRACT Data System Library Handbook for C/ORACLE*.

- ❑ You cannot have units with the same name in a schema, even if the owner names are different.
- ❑ Any filter applied at the source unit or source part level applies to every target unit or target part the source item is mapped to. If you need to work around this limitation, you have the following options:
- Use target unit or target part filters instead of source unit or source part filters
 - Split the conversion into multiple conversions

ORACLE Pro*C Pre-Compiler Limitation

ORACLE's Pro*C pre-compiler limits the number of lines allowed in a Pro*C program to 32,000. When a program exceeds this limit, compiler errors are generated. This is a limitation imposed by the pre-compiler, and *not* by ETI•EXTRACT or the DSL for C/ORACLE. The ORACLE CORPORATION is aware of this limitation and has made available a patch that increases the allowable program size to 64,000 lines. Should you encounter this problem, please contact your ORACLE representative to obtain the patch.

Changes to Templates in Release 1.2.4

Table 2 lists additions to templates in this release. Table 3 lists changes to existing templates in this release.

Table 2. Release 1.2.4 Template Additions

Addition	Description
Added CVAR uow_sql_commit_limit to: <ul style="list-style-type: none"> • uow-status-struct • uow-commit • unitize-check-global-cvars 	Added new template to manage the number of primary key values per commit. Separated the UoW commit limit from the non-UoW sql_commit_limit.
oracle_populate/ora_proc_populate/uow-check-unit-part	Added new template to ensure that all units and parts referenced in the UoW file exist in the target schema.

Table 3. Release 1.2.4 Template Changes

Change	Description
oracle-populate/ora_proc_table_lookup/lookup-function-where-rhs	Changed so that a table lookup comparing a column to zero (0) no longer yields an empty result.
oracle_populate/ora_proc_populate/process-ifile-for-target-unit	Changed to recognize the sql_error_limit value in a conversion where the conversion property data_conv_mode is set to data_driven.
oracle_populate/or_proc_table_lookup/lookup-function-where-rhs	Changed so that a table lookup comparing a column to zero (0) no longer yields an empty result.
oracle_populate/ora_proc_table_lookup/lookup-function-init-hostvar-num-to-str	Changed to correct parenthesis error.
oracle_populate_ora_proc_populate/uow-define-variables	Issues an error if the .ctl file defines a unit or a part that is not a valid target unit or part.
oracle_populate/ora_proc_table_lookup/lookup-function-where-phrase	Changed to correct table lookup with nested subquery.

Changes to DAS Object in Release 1.2.4

There are no changes to the DAS object in Release 1.2.4.

Other Considerations

Other items of consideration that pertain to this release of the DSL for C/ORACLE are as follows:

- ❑ The **RI_dependency_level** context variable is reserved for future use and may contain incorrect values.
- ❑ There must be at least one primary key defined in a retrieved schema. If there are no primary keys defined in a retrieved schema, the following error message is generated when the retrieve-schema Tcl script is executed:

No matching START_ARRAY for FINISH_ARRAY.
- ❑ The retrieve-schema SQL script (oracle_rs.sql) retrieves precision and scale information for numeric data types from the ORACLE data dictionary. The precision and scale information is then used to derive the ETI•EXTRACT data types and lengths for numeric data types. Chapter 2 in the *ETI•EXTRACT Data System Library Handbook for C/ORACLE* shows the precision and scale used to derive the ETI•EXTRACT data types.

