

# VSI OpenVMS

## VSI DECforms Release Notes

**Operating System and Version:** VSI OpenVMS IA-64 Version 8.4-1H1 or higher  
VSI OpenVMS Alpha Version 8.4-2L1 or higher

**Software Version:** DECforms Version 4.0

---

## VSI DECforms Release Notes



VMS Software

---

Copyright © 2024 VMS Software, Inc. (VSI), Boston, Massachusetts, USA

### Legal Notice

Confidential computer software. Valid license from VSI required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for VSI products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. VSI shall not be liable for technical or editorial errors or omissions contained herein.

HPE, HPE Integrity, HPE Alpha, and HPE Proliant are trademarks or registered trademarks of Hewlett Packard Enterprise.

Intel, Itanium and IA-64 are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Motif is a registered trademark of The Open Group.

Oracle is a registered trademark of Oracle and/or its affiliates.

PostScript is a registered trademark of Adobe Systems, Incorporated.

<b>Preface</b> .....	<b>v</b>
1. About VSI .....	v
2. Intended Audience .....	v
3. Operating System Information .....	v
4. Layered Product Dependencies .....	v
5. Related Documents .....	v
6. Acknowledgment .....	vi
<b>Chapter 1. Release Notes</b> .....	<b>1</b>
1.1. DECforms Overview .....	1
1.2. Problems Corrected in Version 4.0 .....	1
1.2.1. Fatal Internal Error with Floating Point Operations on I64 Platform .....	1
1.2.2. FDE ACCVIO Problem during REMOVE TO CLIPBOARD on Alpha Platform .....	1
1.2.3. Forms Extract Object Problem on I64 Platform .....	2
1.2.4. D-float Problem on I64 Platform .....	2
1.2.5. Conditional Handlers do not Work on I64 Platform .....	3
1.2.6. H-float Operations Results in Fatal Error on I64 Platform .....	3
1.2.7. Invalid Comparison Output for H-float on I64 Platform .....	3
1.2.8. ACCVIO Problem in FORMS Application on I64 Platform .....	3
1.2.9. /KEEP Qualifier is Added on I64 Platform .....	4
1.2.10. %LIB-W-EOMWARN, Compilation Warnings in Module FORMS\$TRNSHR .....	4
1.3. Restrictions in this Release .....	4
1.3.1. C Compiler Requirement on I64 Platform .....	4
1.3.2. FORMS CONVERT FMS Command is Removed on I64 Platform .....	4
1.3.3. Linking a Shareable Image .....	5
1.3.4. No DECwindows Support on both Alpha and I64 Platform .....	5
1.3.5. No Web Connector Support .....	5
1.3.6. No support for two Fonts on I64 Platform .....	5
1.3.7. Forms Extract Object does not Generate the List File on I64 Platform .....	5
1.3.8. Other Restrictions in this Release .....	5
1.4. DECforms Portable Bindings for C and FORTRAN .....	6
1.4.1. Placeholder Restriction for FORTRAN .....	6
1.5. Run-Time System (Form Manager) Notes .....	6
1.5.1. General Restrictions .....	6
1.5.1.1. Character-Cell Layout Restrictions .....	8
1.6. Back Translator Notes .....	8
1.6.1. Restrictions in this Release .....	8
1.7. Form Development Environment Notes .....	9
1.7.1. Restrictions in this Release .....	9
1.7.2. Known Problems with Workarounds .....	9
1.8. Character-Cell Panel Editor Notes .....	9
1.8.1. Restrictions in this Release .....	10
1.8.2. Known Problems with Workarounds .....	10
1.9. LSE Support Notes .....	10
1.9.1. Restrictions in this Release .....	10
1.10. Demonstration Forms Notes .....	11
1.10.1. Location of the Demonstration Guide .....	11
1.11. Run-Time Message Translations (DECforms for OpenVMS) .....	12



# Preface

This document contains the release notes for DECforms Version 4.0 for OpenVMS users.

The previous release of DECforms on OpenVMS VAX and Alpha was Version 3.3. This release notes highlight all changes made within DECforms since Version 3.3.

This document often refers to VSI products by their abbreviated names:

- VSI DECforms software is referred to as DECforms.
- DEC Language-Sensitive Editor software is referred to as LSE.
- VSI ACMS software is referred to as ACMS.

## 1. About VSI

VMS Software, Inc. (VSI) is an independent software company licensed by Hewlett Packard Enterprise to develop and support the OpenVMS operating system.

## 2. Intended Audience

This document is intended for all DECforms users.

## 3. Operating System Information

OpenVMS operating system supported:

- OpenVMS Alpha Version 7.3-2 without support for Oracle CDD/Repository Version 7.2 or later
- OpenVMS Alpha Version 8.2 with support for Oracle CDD/Repository Version 7.2 or later
- OpenVMS I64 Version 8.2 without support for Oracle CDD/Repository Version 7.2 or later
- OpenVMS I64 Version 8.2-1 with support for Oracle CDD/Repository version 7.2 or later

## 4. Layered Product Dependencies

To use the Oracle CDD/Repository features found in DECforms Version 4.0, you must use Oracle CDD/Repository Version 7.2 or later.

To use the LSE features found in DECforms Version 4.0, you must use DEC LSE Version 3.0 or later.

## 5. Related Documents

For additional information on using DECforms, refer to the following manuals:

- *VSI DECforms Guide to Commands and Utilities*
- *VSI DECforms Guide to Developing an Application.*

- *VSI DECforms Style Guide for Character-Cell Devices*
- *VSI DECforms IFDL Reference Manual*
- *VSI DECforms Programmer's Reference Manual*
- *VSI DECforms Guide to Converting FMS Applications*
- *VSI DECforms Installation Guide for OpenVMS Systems*

## **6. Acknowledgment**

DECforms is the VSI implementation of a Form Interface Management System (FIMS) ANSI/ISO standard prepared by the CODASYL Form Interface Management System Committee and ISO/IEC JTC1/SC22 Working Group 18. The FIMS standard is documented in the *ISO DIS 11730: December 7, 1992 FIMS* and can be purchased from the International Organization for Standardization or from the American National Standards Institute.

# Chapter 1. Release Notes

## 1.1. DECforms Overview

DECforms integrates text and simple graphics into forms and menus. Application programs use these forms and menus as user interfaces. DECforms also provides extensive facilities for specifying full control of the user interface within the form rather than in the application program.

DECforms aids in application development by separating form and function. The entire user interface is completely external to the application program. The purpose of a form is to separate the application program from the user's view. The program processes the data, and the interface to the user is provided entirely by the form. The form contains terminal management functions, freeing the application program from device dependence.

The programming interface in DECforms is a record-level interface. Data is transferred on a record-by-record basis. Records defined in the program and in the form control how data is transferred between the form and the program.

## 1.2. Problems Corrected in Version 4.0

The following problems have been fixed in Version 4.0:

### 1.2.1. Fatal Internal Error with Floating Point Operations on I64 Platform

#### Problem description

This problem was reported on rx4640, OpenVMS Itanium 8.2, while migrating from VAX environment to Itanium. This fatal internal error (FATINTERR) was reported when the "OUTPUT WHEN" clause was specified on the floating point fields. Whereas, this clause works properly on the integer and character-based fields. F-float to H-float conversion routine invoked through CVTFH macro was the cause for this error.

#### Solution:

This problem was reported in WFM 1206795244-121 and has been fixed in this release.

### 1.2.2. FDE ACCVIO Problem during REMOVE TO CLIPBOARD on Alpha Platform

#### Problem description

Forms Development Environment (FDE) terminates abnormally with ACCVIO error when REMOVE TO CLIP BOARD function is used followed by object selection and DESELECT ALL function.

Steps to reproduce the problem:

- Select two objects and use the "Remove to Clipboard" function. The message is displayed as "2 selected objects removed to the clipboard".

- Select one object. The message is displayed as "The selection list contains 2 objects".
- Type "Deselect All". An access violation error displayed.

**Solution:**

This problem was reported in WFM 3209705942-321 and has been fixed in this release.

### 1.2.3. Forms Extract Object Problem on I64 Platform

**Problem description:**

When an object file is extracted from the .form file, the following error is displayed:

```
$ forms extract object test1
pic, lcl, noshr, noexe, nowrt, novoc, long
.....^
%CC-E-DECLARATOR, Invalid declarator.
at line number 38 in file SYS$LOGIN:TEST1.C;31
%FORMS-E-EXECERR, Error attempting to EXEC a command.
```

**Solution:**

Earlier the intermediate files were getting generated in the same directory where the command was executed.

The problem occurs while generating the newer version of the intermediate file and when the older version still exists in the current directory.

If the expected intermediate files attributes are different from the older one, the preceding error is displayed.

The C Run Time Library will create a new file with the default file attributes if that directory contains no such file. On the other hand, if that directory contains the older version of the intermediate file, then C Run Time Library will create a newer version of the intermediate file but still carry forwards the attributes of the older version. This was causing the problem while generating the intermediate file (both .c and .h).

This problem is fixed by modifying the forms extract object utility. Now the Forms extract object creates a subdirectory named decforms\_temp under the working directory and places the intermediate files extracted from the form file in the same directory.

This problem has been fixed in this release.

### 1.2.4. D-float Problem on I64 Platform

**Problem description:**

All the Negative fractions are converted to 0 upon translation.

**Solution:**

The D-float numbers are converted properly. This problem has been fixed in this release.



## 1.2.5. Conditional Handlers do not Work on I64 Platform

### Problem description:

An event log file does not contain stack dump information for an exception raised. Since the ICB structure is changed from Alpha to I64, the old utility of Alpha does not work on I64.

### Solution:

This problem has been fixed in this release.

## 1.2.6. H-float Operations Results in Fatal Error on I64 Platform

### Problem description:

I64 does not support the arithmetic operation on H-float.

### Solution:

A new routine is provided to covert H-float to IEEE- float and vice-versa. This problem has been fixed in this release.

## 1.2.7. Invalid Comparison Output for H-float on I64 Platform

### Problem description:

H-float comparison failed with unpredictable results.

### Solution:

This problem has been fixed in this release.

## 1.2.8. ACCVIO Problem in FORMS Application on I64 Platform

### Problem description:

DECforms fails with an ACCVIO error for user supplied ifdl file. Following sequence of commands result in ACCVIO error.

```
$ forms translate test_form.ifdl
$ forms extract object test_form
```

The main purpose of `.forms extract object.` is to generate an intermediate `.h` and `.c` files. While generating the header (`.h`) file, DECforms traverses through each line in the form file. For each line it calculates the line number and remaining number of bytes to be copied into header file, and finally output the line into header file. While calculating the remaining number of bytes, DECforms fails with unpredictable result.

## **Solution:**

This problem has been fixed in this release.

### **1.2.9. /KEEP Qualifier is Added on I64 Platform**

#### **Problem description:**

/KEEP qualifier is added for the command FORMS EXTRACT OBJECT to keep the intermediate files generated while creating the object files.

#### **Solution:**

This problem has been fixed in this release.

### **1.2.10. %LIB-W-EOMWARN, Compilation Warnings in Module FORMS\$TRNSHR**

#### **Problem description:**

The command "forms trans" command displayed the following warning message:

```
$ forms trans /list=FORMS$SAMPLE_FORM.IFDL_lis
SYS$COMMON:[SYSHLP.EXAMPLES.FORMS]FORMS$SAMPLE_FORM.IFDL
%LIB-W-EOMWARN, compilation warnings in module FORMS$TRNSHR
```

This problem occurred only on OVMS Dory. FORMS\$TRNSHR.EXE image has a VMS completion code WARNING leading to this error messages.

#### **Solution:**

This problem has been fixed in this release.

## **1.3. Restrictions in this Release**

The following are the restrictions for DECforms Version 4.0.

### **1.3.1. C Compiler Requirement on I64 Platform**

#### **Problem description:**

A C compiler is required to create forms object files. The C compiler must be installed on the same machine where the FORMS EXTRACT OBJECT command is executed.

### **1.3.2. FORMS CONVERT FMS Command is Removed on I64 Platform**

#### **Problem description:**

The FORMS CONVERT FMS command is not supported on DECforms V4.0 for I64 platform.

### 1.3.3. Linking a Shareable Image

#### Problem description:

There is a known issue with linking a shareable image that contains a form, such as described in Section 2.3.5 of the VSI DECforms Programmer's Reference Manual. For example:

```
$ LINK/SHARE/EXE=FORMS$$SAMPLE_FORM FORMS$$SAMPLE_FORM, SYS$INPUT/OPT -  
_ $ SYMBOL_VECTOR=(FORMS$AR_FORM_TABLE=DATA)
```

For the link operation to be successful, the qualifiers /MAP/FULL may be required on the link command line.

### 1.3.4. No DECwindows Support on both Alpha and I64 Platform

#### Problem description:

DECwindows Motif or XWindows displays, including the Motif Panel Editor is not supported on both Alpha and I64 platforms. Several problems have been discovered with DECforms' use of older, unsupported interfaces to the DECwindows product.

### 1.3.5. No Web Connector Support

Web Connector product is not supported on both Alpha and I64 platforms.

### 1.3.6. No support for two Fonts on I64 Platform

#### Problem description:

The CDS\$K\_DEC\_HANGUL and CDS\$K\_DEC\_CNS11643 fonts are not supported on I64. When forms containing these font specifiers are translated, the default font is used, and a warning message is issued.

### 1.3.7. Forms Extract Object does not Generate the List File on I64 Platform

#### Problem description:

The FORMS EXTRACT OBJECT command will extract the object file from the .FORM file. DECforms version 4.0 does not generate the list file with /LIST qualifier while extracting the object file.

### 1.3.8. Other Restrictions in this Release

#### Problem description:

It is recommended that you avoid using layouts which have UNITS POINTS specified. Although UNITS POINTS translates as valid IFDL syntax, unpredictable results may occur if you use layouts with UNITS POINTS specified. Note that this restriction will be lifted in a future release of DECforms.

## 1.4. DECforms Portable Bindings for C and FORTRAN

This section documents new features, known restrictions, and problems in the DECforms Version 4.0 Portable API.

### 1.4.1. Placeholder Restriction for FORTRAN

When specifying missing string arguments in FORTRAN bindings, you must use one of the following placeholders:

```
forms_enable_for (session_id, ,  
                 form_file, form_name, request_option)
```

```
forms_enable_for (session_id, "",  
                 form_file, form_name, request_option)
```

```
forms_enable_for (session_id, CHAR(0),  
                 form_file, form_name, request_option)
```

Specifying missing string arguments using the numeric value, 0, is considered a programming error because 0 is of type integer. DECforms does not guarantee detection of such errors. These errors may produce unexpected behavior at runtime.

In addition, when a DECforms request call is expecting *n* incoming user parameters, where *n* is a positive integer, you must specify each parameter either by passing an actual variable, value, or placeholder. For example, the `forms_enable` call expects five incoming parameters, as follows:

```
forms_enable_for (session_id,, form_file,,) ❶  
forms_enable_for (session_id,, form_file,)  ❷
```

- ❶ This `forms_enable` call is correct. The five parameters are:

```
session_id  
,  
  
form_file  
,  
,
```

- ❷ This `forms_enable` call is incorrect.

## 1.5. Run-Time System (Form Manager) Notes

This section documents known restrictions and problems in the DECforms Version 4.0 Form Manager.

### 1.5.1. General Restrictions

- When a DECforms application written in C is compiled with the `/EXTERNAL_MODEL=COMMON_BLOCK` parameter, the application could fail at run-time with a "no read access to user argument" error.
- Asynchronous requests cannot be used when DECThreads threading is active in the process. The error `FORMS$_BLOCKED_BY_THREAD` is returned by any attempt to perform such a request. You can correct this situation in one of two ways:

1. Remove DECthreads threading from the process
2. Recode so that the current AST request is done from a thread

DECforms has added this restriction because it has always been a DECthreads restriction and we cannot handle resolution of customer problems without satisfying this DECthreads requirement.

- Images linked using DECforms Version 4.0 will not operate against previous versions of DECforms and will return shareable image ident mismatch when attempted.
- DECforms sets the `tm_wday` and `tm_yday` fields to 0 in the `FORMS_TM` structure before passing it back to the user application. If that application needs to use these fields, the fields must be set properly. Use the C RTL `mktime()` function, which sets the `tm_wday` and `tm_yday` fields as a side effect, to do this.

Beginning with OpenVMS V7.0, the C RTL `asctime()` function (used to convert a struct `tm` to an ASCII string), uses the `tm_wday` field to correctly compute the day-of-the-week portion of the output string. A value of 0 in `tm_wday` always results in an output of Sunday. On pre-V7.0 systems, `asctime()` ignored the `tm_wday` field and computed the day-of-the-week from the other fields in the struct `tm`.

- The form record field offsets and lengths are calculated at run time on OpenVMS Alpha when loading a `.FORM` file. This allows the `.FORM` file to be compatible between the OpenVMS VAX and OpenVMS Alpha platforms.
- The form data structures are new as of DECforms Version 2.1. Old Versions of the form data structures created before Version 2.1 that are contained in `.FORM` files are converted when the form is loaded (during the `FORMS$ENABLE` call). Previous versions of the form data structures contained in images and from extracted objects are converted and reloaded from the old copy of the data structure. A new copy is made in memory. For performance reasons the `.FORM` files should be recreated using the Version 2.1 or 2.2 IFDL Translator. Extracted objects should be reextracted using the Version 2.1 or 2.2 Extract Objects Utility, and images should be rebuilt under DECforms Version 2.1 or 2.2.

- UNITS POINTS

Avoid using layouts that specify UNITS POINTS. Using layouts with UNITS POINTS may result in unpredictable results. This restriction may be lifted in a future release of DECforms.

- Form data items and multiple panel fields.

A single form data item can be displayed in multiple panel fields. However, either the picture strings specified for those panel fields should allow any value that the data item may contain to be displayed, or the value should be set to a value that can be displayed in the panel prior to displaying the panel.

For example, you have a form data item with a data type of Longword Integer, and this data item appears in two panel fields with pictures of 9999999999 and 999, respectively. If the value of the data item is greater than or equal to 1,000, the second panel field's picture cannot display the data. Prior to displaying the second panel, the data item has to be set to a value that is compatible with the 999 picture or an error is returned to the application program.

- Tracing is local when turned on within an escape routine.

When no logical name or request specifies tracing within a form session, the default for tracing is off. In general, tracing remains on or off until a DCL logical or an item code in a request is specified to

change it. However, if the status of tracing is changed within an escape routine, that change of status remains only for the duration of the escape routine execution.

For example, if tracing was off in a form session, but was later turned on within an escape routine, only the events that took place within the escape routine are recorded. Similarly, if tracing was on in a form session, but was later turned off within an escape routine, all events that occurred inside this escape routine are excluded from the trace file.

- RETURN/RETURN IMMEDIATE prior to accept phase.

In DECforms Version 4.0, RETURN and RETURN IMMEDIATE response steps are ignored before entering accept phase if the activation list is not empty. If one of these response steps is in an external response, such as ENABLE or SEND, and some items are activated before the response steps, the response steps will not be executed.

- Under some circumstances, if you are running ACMS in debug mode with DECforms and Oracle TRACE, you may experience failures. As a workaround, define the logical EPC\$SHR to "NL:".

### 1.5.1.1. Character-Cell Layout Restrictions

- CONCEALED WHEN and overlapping panel objects.

If a panel field in a character cell layout contains a CONCEALED WHEN clause, and that panel field overlaps background literals on the panel, the literals do not appear when the CONCEALED WHEN condition is true. Space characters are written in the character positions occupied by the concealed object.

- REFRESH ALL and foreign screen agents.

The REFRESH ALL response step in a character cell layout may sometimes interfere with the operation of other screen packages that may share the screen with DECforms. In addition, if other screen agents are sharing the display with DECforms, REFRESH ALL may not restore the state of the DECforms display.

## 1.6. Back Translator Notes

This section documents known restrictions and problems in the DECforms Version 4.0 Back Translator.

### 1.6.1. Restrictions in this Release

- 

- Rectangle and polyline literals in absolute coordinates

When the Back Translator outputs the line and column clauses for rectangles and polylines, it outputs these clauses using absolute coordinates regardless of how they were specified previous to translation.

- Comments may be moved between translation and back translation.

The IFDL Translator creates a compact binary representation of your form that can be enabled by the Form Manager and manipulated efficiently by the Panel Editor and the Form Development Environment. To make the structure efficient to manipulate, the Translator cannot precisely record

the location of comments, because a comment can be placed between any two syntax tokens. Instead, the location of a comment is approximated in the form file, so when the Back Translator reconstructs the IFDL file from the form file, it may not place your comment exactly where it was in the original file.

Once an IFDL file has been written by the Back Translator, the comments are in positions from which they will not move in repeated passes through the Translator and Back Translator. An effective workaround is to let the comments move once, and then modify them so that they are meaningful at their new locations. If this is not adequate, the next best workaround is to avoid using the Back Translator, which means avoiding the Panel Editor and Form Development Environment.

By refraining from the use of the Panel Editor and Form Development Environment you are using IFDL as an ordinary text-based language, with a compiler and an execution component.

- COPY format 1 statements are lost between translation and back translation.

Because a COPY format 1 statement can appear anywhere in an IFDL file that a comment can appear, the Translator has the same problem with COPY statements that it has with comments. However, moving a COPY statement can change the meaning of an IFDL file, so rather than take the chance of creating an incorrect IFDL file, the COPY statements are replaced by a back translation of the information copied.

If the loss of the COPY statement is unacceptable, the only useful workaround is to avoid using the Back Translator, which means avoiding the Panel Editor and the Form Development Environment.

## 1.7. Form Development Environment Notes

This section documents known restrictions and problems in the DECforms Version 4.0 Form Development Environment (FDE).

### 1.7.1. Restrictions in this Release

Overwritten error messages

The display of hints overwrites the display of error messages. When you press PF1-PF2, the automatic display of Help messages (hints) is turned on. These hints occasionally overwrite error messages, such as "failed validation".

### 1.7.2. Known Problems with Workarounds

Terminal width change problem within the FDE If you use the TEST or EDIT PANEL options within the FDE and the panel being used is of a different terminal width than the main FDE session, the terminal width is not reset to the original width after you exit either option and return to the main FDE menu.

The terminal width is reset appropriately when you exit the FDE.

## 1.8. Character-Cell Panel Editor Notes

This section documents known restrictions and problems in the DECforms Version 4.0 Character-Cell Panel Editor.

## 1.8.1. Restrictions in this Release

- Panels associated with viewports declared FOR PRINTING cannot be edited by the Character-Cell Panel Editor (CCPED) unless both the terminal/DECterm size and the default viewport size (the layout's SIZE declaration) are large enough to accommodate the panels.
- The command recall buffer stops recording commands after the 16th command. The recall buffer does not lose the previous 16 commands, and they can be recalled, but after the 16th one, no more commands will be recorded.

## 1.8.2. Known Problems with Workarounds

When certain field description attributes are changed from a conditional to an unconditional format of the attribute (possible with PROTECTED/PROTECTED WHEN and CONCEALED/CONCEALED WHEN), the conditional clause is not always properly erased from the internal representation of the form. This does not affect the run-time operation of the form. The Form Manager correctly sees the unconditional case. However, if you remove the unconditional attribute before translating the form file back into an IFDL source file, and then translate that source file into a form file, the conditional attribute may reappear.

You can prevent the reappearance of a conditional clause you have removed in the Character-Cell Panel Editor by translating the form file into an IFDL source file and then translating the IFDL source file into a form file again. These translations purge the unused conditional clause.

## 1.9. LSE Support Notes

This section documents new features, known restrictions and problems in the DECforms Version 4.0 Language-Sensitive Editor Support.

### 1.9.1. Restrictions in this Release

- Avoid using layouts that specify UNITS POINTS. Although UNITS POINTS is included as valid IFDL syntax, unpredictable results may occur if you use layouts with UNITS POINTS specified. This restriction may be lifted in a future release of DECforms.
- The DECforms language support package supplies a placeholder titled REC\_ARGS in the FORMS\$SEND, FORMS\$RECEIVE, and FORMS\$TRANSCEIVE calls. This placeholder replaces all instances of the placeholders Send\_Record\_Message, Send\_Shadow\_Record, Receive\_Record\_Message, and Receive\_Shadow\_Record. The DECforms send, receive and transeive entry points may accept a variable number of actual parameters for the formal parameters Send\_Record\_Message, Send\_Shadow\_Record, Receive\_Record\_Message, and Receive\_Shadow\_Record.

However, it is not possible to implement the Forms\$Entry\_Points package in a way that would generate multiple occurrences of these parameters without special processing. REC\_ARGS is intended to denote data record and shadow record parameter pairs. In the case of FORMS\$SEND, REC\_ARGS denotes an instance of the send data and send shadow record pair. In the case of FORMS\$RECEIVE, REC\_ARGS denotes an instance of the receive data and receive shadow record pair. In the case of FORMS\$TRANSCEIVE, REC\_ARGS denotes an instance of both the send and receive record pairs described above. Please refer to the VSI DECforms Programmer's Reference Manual for exact details on the calling format for the FORMS\$SEND, FORMS\$RECEIVE, and FORMS\$TRANSCEIVE entry points.



- If you are editing a Pascal buffer in LSE, and you use the Forms\$Entry\_Points package to generate DECforms calls, note that the keyword syntax generated for FORMS\$SEND, FORMS\$RECEIVE, and FORMS\$TRANSCIVE will need to be changed to positional syntax. For example, the Language-Sensitive Editor generates the following syntax for the FORMS\$RECEIVE call:

```
forms$receive (
    session_id := {%session_id.d}%,
    receive_name := {%receive_name.d}%,
    receive_count := {%receive_count.r}%,
    [%receive_ctext := {%receive_ctext.d}%,
    [%rec_ctext_ct := {%rec_ctext_ct.r}%,
    [%send_ctext := {%send_ctext.d}%,
    [%send_ctl_cnt := {%send_ctl_cnt.r}%,
    [%timeout := {%timeout.r}%,
    [%parent_id := {%parent_id.d}%,
    [%request_opt := {%request_opt.r}%,
    [%rec_args := {%rec_args}%,
    ]%)
```

The keyword syntax should be changed to positional syntax. The following FORMS\$RECEIVE call shows the correct positional format:

```
forms$receive(
    session_id,           { session id }
    'account',           { record name in form }
    ,                   { Number of recs sent, default 1 }
    receive_ctl_txt::char25, { Receive ctl text msg }
    receive_ctl_txt_ct,   { Receive ctl text count }
    , ,                 { Send ctl text msg/count }
    ,                   { timeout }
    ,                   { parent request id }
    ,                   { request options item list }
    account_temp,       { the record }
    );                  { shadow record }
```

The argument REC\_ARGS is assigned the "LIST" attribute in the DECforms Pascal definitions file. This attribute permits the passing of multiple actual parameters in the space of one formal parameter. Because of the nature of the FORMS\$SEND, FORMS\$RECEIVE, and FORMS\$TRANSCIVE calls, it may be necessary to pass more than one parameter in the REC\_ARGS argument (a data record and a shadow record, for example). However, keyword syntax does not support multiple actual arguments for one formal argument.

## 1.10. Demonstration Forms Notes

This section documents changes in the DECforms Version 4.0 Demonstration Package.

### 1.10.1. Location of the Demonstration Guide

The Demonstration Guide is available in both PostScript and ASCII versions:

```
FORMS$EXAMPLES:FORMS$DEMO_GUIDE.PS
FORMS$EXAMPLES:FORMS$DEMO_GUIDE.TXT
```

You may print either of these files on an appropriate printer for complete details about the demonstration package.

## 1.11. Run-Time Message Translations (DECforms for OpenVMS)

Included in DECforms for OpenVMS is the ability for form developers to manually translate run-time messages into languages other than English. For detailed information on how to use this feature, install the modifiable run-time message support (an installation option), and read the following documentation:

```
SYS$SYSROOT:[SYSHLP.EXAMPLES.FORMS.MESSAGES]DECFORMS_MESSAGES.TXT
```