

**ELIST.8100.Final.SOL7.STP\_STD\_STR**

**Defense Information Infrastructure (DII)  
Common Operating Environment (COE)**

**Software Test Plan/Description/Report (STP/STD/STR)  
for the  
Enhanced Logistics Intratheater Support Tool (ELIST)  
Global Data Segment Version 8.1.0.0,  
Database Instance Segment Version 8.1.0.0,  
Database Fill Segment Version 8.1.0.0,  
Database Segment Version 8.1.0.0,  
Database Utility Segment Version 8.1.0.0,  
Software Segment Version 8.1.0.0, and  
Reference Data Segment Version 8.1.0.0**

**for Solaris 7**

**26 February 2002**

**Prepared for:**

**Military Traffic Management Command  
Transportation Engineering Agency  
720 Thimble Shoals Boulevard  
Newport News, VA 23606**

**Prepared by:**

**Argonne National Laboratory  
Decision and Information Sciences Division  
9700 South Cass Avenue  
Argonne, IL 60439**



## Table of Contents

1. Scope.....	1
1.1 Identification.....	1
2. Referenced Documents.....	3
2.1 Government Documents.....	3
2.1.1 DII COE ELIST Documents.....	3
2.1.2 Other DII COE Documents.....	3
2.1.3 Other ELIST Documents.....	3
2.1.4 Other Government Documents.....	3
2.2 Non-Government Documents.....	3
3. ELIST Software Test Plan.....	5
3.1 Scope of the Testing Effort.....	5
3.2 Testing Roles.....	5
3.3 Testing Methodology.....	5
3.3.1 Procedures for Installation Testing.....	6
3.3.2 Procedures for Functional Testing.....	7
3.3.3 Procedures for Deinstallation Testing.....	7
3.3.4 Summary of Tests Performed.....	8
3.4 Test Environment.....	13
3.4.1 Hardware.....	13
3.4.2 Software.....	14
3.4.3 Operating System Kernel Parameters.....	15
4. Test Descriptions and Results.....	17
4.1 Conventions.....	17
4.1.1 Prerequisite Conditions.....	17
4.1.2 Test Inputs and Procedures.....	17
4.1.3 Expected Test Results and Criteria for Evaluating Test Results.....	17
4.1.4 Order of the Tests.....	17
4.2 Installation Tests and Results.....	17
4.2.1 ELIST Global Data Segment.....	17
4.2.2 ELIST Database Instance Segment.....	19
4.2.3 ELIST Database Fill Segment.....	30
4.2.4 ELIST Database Segment.....	31
4.2.5 ELIST Database Utility Segment.....	35
4.2.6 ELIST Software Segment.....	36
4.2.7 ELIST Reference Data Segment.....	38
4.3 Functional Tests of the Features of the ELIST Database Utility Segment.....	40
4.3.1 Create User Account Feature.....	40
4.3.2 Remove User Account Feature.....	46
4.3.3 Lock User Account Feature.....	51
4.3.4 Unlock User Account Feature.....	55
4.3.5 List User Accounts Feature.....	59
4.3.6 Change User Password Feature.....	60
4.4 Functional Tests of the Administrative Features of the ELIST Software Segment.....	65
4.4.1 Add Map Data Feature.....	65

4.4.2	Delete Map Data Feature .....	71
4.5	Functional Tests of the General Features of the ELIST Software Segment.....	74
4.5.1	Run ELIST Feature.....	75
4.5.2	Run ETEdit Feature .....	78
4.6	Deinstallation Tests and Results .....	81
4.6.1	ELIST Reference Data Segment.....	81
4.6.2	ELIST Software Segment .....	83
4.6.3	ELIST Database Utility Segment .....	85
4.6.4	ELIST Database Segment.....	86
4.6.5	ELIST Database Fill Segment .....	89
4.6.6	ELIST Database Instance Segment.....	90
4.6.7	ELIST Global Data Segment .....	93
5.	Notes .....	95
6.	Acknowledgements.....	97
7.	Documentation Improvement and Feedback .....	99

## List of Tables

Table 1. Segments of the ELIST Mission Application .....	1
Table 2. Summary of Installation Tests .....	8
Table 3. Summary of Functional Tests of the Features of the ELIST Database Utility Segment	9
Table 4. Summary of Functional Tests of the Administrative Features of the ELIST Software Segment.....	11
Table 5. Summary of Functional Tests of the General Features of the ELIST Software Segment .....	12
Table 6. Summary of Deinstallation Tests.....	12
Table 7. Size of Disk Partitions on the Testbed Machines .....	14
Table 8. Infrastructure Service Segments Required by ELIST .....	15
Table 9. Solaris Kernel Parameter Values on the Database Server .....	15

## List of Figures

Figure 1. Interactions among the Test Participants.....	6
Figure 2. Segment Installer Window after Installing the ELIST Global Data Segment .....	18
Figure 3. Window for Expressing the Principal Choice While Installing the ELIST Database Instance Segment .....	20
Figure 4. Window for Entering a New Instance Name When Instances Already Exist .....	21
Figure 5. Window Following Failure to Enter a New Instance Name .....	21
Figure 6. Message Informing that Database Instance Exists.....	22
Figure 7. Window Prompting for Password to be Assigned to SYSTEM User of the New Database Instance .....	22
Figure 8. Message Noting Failure to Assign SYSTEM Password.....	23
Figure 9. Message Noting Successful Creation of New Database Instance .....	24
Figure 10. Segment Installer Window after Installing the ELIST Database Instance Segment.....	25
Figure 11. Terminal Window Showing that Instance Directories and Files Exist .....	26
Figure 12. Window for Entering an Existing Instance Name.....	27
Figure 13. Window Following Failure to Enter an Existing Instance Name.....	27
Figure 14. Message Informing that Database Instance Does not Exist .....	28
Figure 15. Window Reminding that Existing Instance Will Persist after Future Segment Deinstallation .....	28
Figure 16. Window for Entering a New Instance Name When No Instances Exist .....	30
Figure 17. Segment Installer Window after Installing the ELIST Database Fill Segment.....	31
Figure 18. Prompt for SYSTEM Password.....	32
Figure 19. Message Noting Failure to Satisfy Prompt for SYSTEM Password .....	32
Figure 20. Message Noting Successful Creation of the ELIST Database .....	33
Figure 21. Segment Installer Window after Installing the ELIST Database Segment .....	34
Figure 22. Segment Installer Window after Installing the ELIST Database Utility Segment.....	35
Figure 23. Segment Installer Window after Installing the ELIST Software Segment on the Database Server .....	37
Figure 24. Segment Installer Window after Installing the ELIST Software Segment on the Application Client.....	38
Figure 25. Segment Installer Window after Installing the ELIST Reference Data Segment on the Database Server .....	39
Figure 26. Segment Installer Window after Installing the ELIST Reference Data Segment on the Database Server .....	40
Figure 27. Window Prompting for the Database Account to Create .....	41
Figure 28. Window Confirming Abort of Create User Account .....	41
Figure 29. Window Prompting for the Password to be Assigned to the New Database Account.....	42
Figure 30. Window Confirming Failure to Assign a Password to the New Database Account .....	43
Figure 31. Window Prompting for the SYSTEM Password of the ELIST Database Instance .....	43
Figure 32. Window Confirming Failure to Enter the SYSTEM Password of the ELIST Database Instance .....	44
Figure 33. Window Confirming that the New Database Account Has Been Created .....	45
Figure 34. SQL*Plus Window Showing that the Database Account Exists and Has the Correct Password .....	45
Figure 35. Window Reporting that the Database Account Already Exists .....	46

Figure 36. Window Prompting for the Database Account to Remove .....	47
Figure 37. Window Confirming Abort of Remove User Account.....	47
Figure 38. Window Confirming Failure to Enter the SYSTEM Password of the ELIST Database Instance .....	48
Figure 39. Window Confirming that the Database Account Has Been Removed.....	49
Figure 40. SQL*Plus Window Showing that the Database Account No Longer Exists .....	50
Figure 41. Window Reporting that the Database Account Does Not Exist.....	51
Figure 42. Window Prompting for the Database Account to Lock .....	51
Figure 43. Window Confirming Abort of Lock User Account.....	52
Figure 44. Window Confirming Failure to Enter the SYSTEM Password of the ELIST Database Instance .....	53
Figure 45. Window Confirming that the Database Account Has Been Locked .....	53
Figure 46. SQL*Plus Window Showing that the Database Account Has Been Locked .....	54
Figure 47. Window Prompting for the Database Account to Unlock.....	55
Figure 48. Window Confirming Abort of Unlock User Account .....	56
Figure 49. Window Confirming Failure to Enter the SYSTEM Password of the ELIST Database Instance .....	57
Figure 50. Window Confirming that the Database Account Has Been Unlocked.....	57
Figure 51. SQL*Plus Window Showing that the Database Account Has Been Unlocked .....	58
Figure 52. Window Confirming Failure to Enter the SYSTEM Password of the ELIST Database Instance .....	60
Figure 53. Window Showing the Existing Database Accounts .....	60
Figure 54. Window Prompting for the Database Account Whose Password is to be Changed .	61
Figure 55. Window Confirming Abort of Change User Password.....	61
Figure 56. Window Prompting for the New Password to be Assigned to the Database Account	62
Figure 57. Window Confirming Failure to Assign a New Password to the Database Account .	63
Figure 58. Window Confirming Failure to Enter the SYSTEM Password of the ELIST Database Instance .....	64
Figure 59. Window Confirming that the Password of the Database Account Has Been Changed .....	64
Figure 60. Add Map Data Window with Instructions for Proceeding.....	66
Figure 61. Add Map Data Window after Clicking Cancel .....	67
Figure 62. File Manager Window that Opens after a NIMA Map CD is Inserted into the CDROM Drive.....	68
Figure 63. Add Map Data Window Confirming Adding of Map Data.....	68
Figure 64. File Manager Window Showing the Added Map Data .....	69
Figure 65. Add Map Data Window Indicating that Map Data are Already Online.....	69
Figure 66. Add Map Data Window Indicating Insufficient Space .....	70
Figure 67. Delete Map Data Window Showing the Online Map Data Sets .....	71
Figure 68. Delete Map Data Window after Selecting No Data to Delete.....	72
Figure 69. Delete Map Data Window after Specifying Nonexistent Data to Delete.....	72
Figure 70. Delete Map Data Window after Deleting Map Data.....	73
Figure 71. File Manager Window Confirming the Deletion of Map Data .....	73
Figure 72. Delete Map Data Window when No Map Data Sets Exist.....	74
Figure 73. Application Manager – DII_APPS Window .....	75
Figure 74. Application Manager – ELISTexec Window .....	75
Figure 75. Prompt for the User’s Database Password .....	76
Figure 76. ELIST Logo Window .....	76
Figure 77. ELIST Map Window .....	77
Figure 78. Confirm Termination of ELIST Window.....	78

Figure 79. Window Affording an Opportunity to Pause Before Closing the ELIST Terminal Window ..... 78

Figure 80. ETEdit Logo Window ..... 79

Figure 81. Confirm Termination of ETEdit Window ..... 80

Figure 82. Window Affording an Opportunity to Pause Before Closing the ETEdit Terminal Window ..... 80

Figure 83. Segment Installer Window after Deinstalling the ELIST Reference Data Segment on the Database Server ..... 82

Figure 84. Segment Installer Window after Deinstalling the ELIST Reference Data Segment on the Application Client ..... 83

Figure 85. Segment Installer Window after Deinstalling the ELIST Software Segment on the Database Server ..... 84

Figure 86. Segment Installer Window after Deinstalling the ELIST Software Segment on the Application Client ..... 85

Figure 87. Segment Installer Window after Deinstalling the ELIST Database Utility Segment ..... 86

Figure 88. Message Noting Failure to Satisfy Prompt for SYSTEM Password ..... 87

Figure 89. Message Noting Successful Removal of the ELIST Database ..... 88

Figure 90. Segment Installer Window after Deinstalling the ELIST Database Segment ..... 89

Figure 91. Segment Installer Window after Deinstalling the ELIST Database Fill Segment ..... 90

Figure 92. Message Noting Successful Removal of the ELIST Database Instance ..... 91

Figure 93. Segment Installer Window after Deinstalling the ELIST Database Instance Segment ..... 92

Figure 94. Terminal Window Showing that Instance Directories and Files Exist ..... 92

Figure 95. Message Noting Retention of the Preexisting ELIST Database Instance ..... 93

Figure 96. Segment Installer Window after Deinstalling the ELIST Global Data Segment ..... 94

# 1. Scope

This document is the *Software Test Plan/Description/Report (STP/STD/STR)* for the DII COE *Enhanced Logistics Intratheater Support Tool (ELIST)* mission application. It combines in one document the information normally presented separately in a Software Test Plan, a Software Test Description, and a Software Test Report; it also presents this information in one place for all the segments of the ELIST mission application.

The primary purpose of this document is to show that ELIST has been tested by the developer and found, by that testing, to install, deinstall, and work properly. The information presented here is detailed enough to allow the reader to repeat the testing independently.

The remainder of this document is organized as follows. Section 1.1 identifies the ELIST mission application. Section 2 is the list of all documents referenced in this document. Section 3, the Software Test Plan, outlines the testing methodology and scope—the latter by way of a concise summary of the tests performed. Section 4 presents detailed descriptions of the tests, along with the expected and observed results; that section therefore combines the information normally found in a Software Test Description and a Software Test Report. The remaining small sections present supplementary information.

Throughout this document, the phrase *ELIST IP* refers to the *Installation Procedures (IP) for the Enhanced Logistics Intratheater Support Tool (ELIST) Global Data Segment, Database Instance Segment, Database Fill Segment, Database Segment, Database Utility Segment, Software Segment, and Reference Data Segment*.

## 1.1 Identification

Table 1 identifies all the segments of the ELIST mission application. In the table, each segment is given a number by which it may be referenced in this document. The table also gives the name, the segment type (and, if a data segment, the segment scope), the current version number, and the directory name assigned to each segment.

**Table 1. Segments of the ELIST Mission Application**

Segment Number	Segment Name	Segment Type / Scope	Version Number	Directory Name
1	ELIST Global Data Segment	Data / Global	8.1.0.0	ELISTglob
2	ELIST Database Instance Segment	Data / Segment	8.1.0.0	ELISTdbinst
3	ELIST Database Fill Segment	Data / Local	8.1.0.0	ELISTdbfill
4	ELIST Database Segment	Database	8.1.0.0	ELISTdb
5	ELIST Database Utility Segment	Software	8.1.0.0	ELISTdbutil
6	ELIST Software Segment	Software	8.1.0.0	ELISTexec
7	ELIST Reference Data Segment	Data / Local	8.1.0.0	ELISTrefdata

All seven segments have the following identification properties in common:

<b>Segment Prefix</b> <sup>1</sup> :	ELIST
<b>Platform(s)</b> <sup>2</sup> :	Sun/Solaris 7
<b>DII COE Versions:</b>	4.2.0.0P4 or later

All seven of the ELIST segments must be installed before you can use the ELIST mission application.<sup>3</sup>

Refer to the *Introduction to the Enhanced Logistics Intratheater Support Tool (ELIST) Mission Application and its Segments: Global Data Segment, Database Instance Segment, Database Fill Segment, Database Segment, Database Utility Segment, Software Segment, and Reference Data Segment* for the following:

- an overview of the mission application and all of its segments in the context of the application;
- the definitions of key concepts and terms used throughout the ELIST documentation;
- a complete list of the available ELIST documentation.
- a brief history of ELIST; and
- basic information pertinent to the client/server configuration and installation of the ELIST segments.

---

<sup>1</sup> Note carefully that all segments have the same prefix. This is not typical of multisegment DII COE mission applications.

<sup>2</sup> Implementation of the ELIST segments for PC/Windows NT 4.0 will follow shortly. This documentation covers only the Sun/Solaris 7 platform but will be supplemented or replaced when an implementation becomes available for NT.

<sup>3</sup>To save space, however, the ELIST Database Fill Segment can be removed after successfully installing the ELIST Database Segment.

## 2. Referenced Documents

The following other documents are referenced in this document.

### 2.1 Government Documents

#### 2.1.1 DII COE ELIST Documents

*Introduction to the Enhanced Logistics Intratheater Support Tool (ELIST) Mission Application and its Segments: Global Data Segment Version 8.1.0.0, Database Instance Segment Version 8.1.0.0, Database Fill Segment Version 8.1.0.0, Database Segment Version 8.1.0.0, Database Utility Segment Version 8.1.0.0, Software Segment Version 8.1.0.0, and Reference Data Segment Version 8.1.0.0 for Solaris 7*, ELIST.8100.Final.SOL7.Intro, Argonne National Laboratory, 26 February 2002

*Installation Procedures (IP) for the Enhanced Logistics Intratheater Support Tool (ELIST) Global Data Segment Version 8.1.0.0, Database Instance Segment Version 8.1.0.0, Database Fill Segment Version 8.1.0.0, Database Segment Version 8.1.0.0, Database Utility Segment Version 8.1.0.0, Software Segment Version 8.1.0.0, and Reference Data Segment Version 8.1.0.0 for Solaris 7*, ELIST.1000.Final.SOL7.IP, Argonne National Laboratory, 26 February 2002

*System Administrator's Manual (SAM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Instance Segment Version 8.1.0.0 for Solaris 7*, ELISTdbinst.8100.Final.SOL7.SAM, Argonne National Laboratory, 26 February 2002

#### 2.1.2 Other DII COE Documents

*Installation Procedures (IP) for ORACLE Client Applications (ORAC) Version 2.1.0.0/8.1.6 for Solaris 7*, CM Number 34865, FGM Inc., 15 June 2000

*System Administrator's Manual (SAM) for ORACLE Client Applications (ORAC) Version 2.1.0.0/8.1.6, ORACLE DataBase Administration (ORADBA) Version 2.1.0.0/2.1.0, ORACLE DataBase Instance (ORADBI) Version 2.1.0.0/8.1.6, and ORACLE RDBMS (ORAS) Version 2.1.0.0/8.1.6.1 for Solaris 7*, CM Number 34815, FGM Inc., 15 June 2000

#### 2.1.3 Other ELIST Documents

N/A.

#### 2.1.4 Other Government Documents

N/A.

### 2.2 Non-Government Documents

N/A.

This page left intentionally blank.

### 3. ELIST Software Test Plan

#### 3.1 Scope of the Testing Effort

The installation and deinstallation of all seven segments of the ELIST mission application were tested to ensure that they proceeded smoothly and correctly and that the applicable DII COE requirements were satisfied. This phase of the testing was exhaustive enough to exercise all important options that the administrator can exercise during installation or deinstallation. It also covered both correct and incorrect responses (*i.e.*, out-of-bounds input) on the part of the administrator, thereby testing the robustness of the processing performed by the installation and deinstallation scripts. Particular attention was also paid to the usefulness of the messages displayed by the graphical user interface (GUI) of these scripts, to ensure that they adequately guided the administrator during the installation or deinstallation process.

Basic functional testing was also performed on the features encapsulated in the two software segments, the ELIST Database Utility Segment and the ELIST Software Segment. This testing was sufficient to verify that the features are usable following their installation.

Typically, each test verifies that one aspect of the software being tested works correctly. For example, an individual test might verify that a particular prompt, appropriate to the current state, is displayed for the user. The correctness of the software being tested is established only after performing a series of small tests. Currently, 105 such tests are described in this document.

**NOTE:** Time and resource availability have limited the testing that has been performed to date. Tests beyond those summarized in Section 3.3.4 have not yet been designed or performed; they will be included in later versions of this document.

#### 3.2 Testing Roles

Testing was performed by the DII COE Test Team of the developer, Argonne National Laboratory (ANL), at their site. Other participants were the DII COE Integrator at ANL and, in some cases, the ELIST Development Team at ANL.

ELIST was implemented by the Development Team well before the decision was made to seek DII COE compliance certification; they also performed internal testing concurrent with code development as well as additional functional testing after development. The role of the Test Team was limited to testing; its members were not involved in the development of ELIST or in its segmentation, but they also assisted the Development Team in the post-implementation functional testing. The DII COE Integrator's primary role was the segmentation of ELIST; secondarily, the Integrator also assisted with testing, at least with respect to evaluating test results and determining where problem fixes were needed.

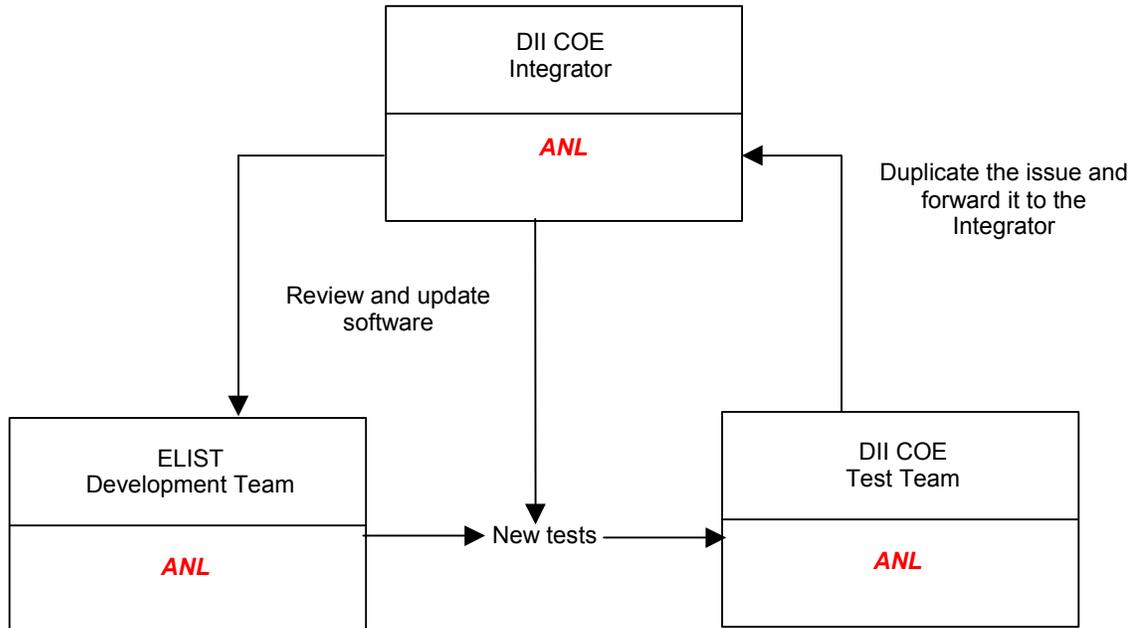
#### 3.3 Testing Methodology

Errors uncovered by the Test Team were first reproduced, then discussed with the DII COE Integrator. Errors in the installation or deinstallation scripts, or in the administrative features of the ELIST Database Utility Segment or the ELIST Software Segment, were fixed by the DII COE Integrator, while errors in the general features of the ELIST Software Segment (*i.e.*, in the application code for ELIST or ETEdit) were referred to the ELIST Development Team for

correction. After the software was corrected, the tests that originally uncovered the error were repeated by the Test Team.

The Test Team and DII COE Integrator have documented in Section 4 of this report the steps involved in each of the tests, the expected results, and the observed results (usually by performing screen captures to show the system response).

Figure 1 shows the feedback that took place during interactions among the DII COE Test Team, DII COE Integrator, and ELIST Development Team.



**Figure 1. Interactions among the Test Participants**

### 3.3.1 Procedures for Installation Testing

The procedures followed during installation testing are summarized as follows:

- Following DII COE rules, the Segment Installer was executed by the system administrator (`sysadmin` account).
- The Segment Installer was invoked from the command line in a terminal window, so as to see non-GUI progress and error messages that would have been missed had it been invoked in the usual GUI manner.
- The Segment Installer was the only application running.
- The installation medium was the disk file representing the `tar` output from running `MakeInstall` at the end of the development process.
- Where relevant, segments were installed on both a database server and an application client; that is, the installation tests were repeated on both types of platforms.

- Erroneous output, if any, was captured.
- Attempts to install segments out of order were originally intended to be tested but were eventually abandoned. The Segment Installer uses the `REQUIRES` segment descriptor to enforce the installation order, but it is also capable of implicitly installing prerequisite segments if they are on the installation medium but are not yet installed. Unfortunately, at the present time (that is, in COE Version 4.2.0.OP4), the Segment Installer does not compute the *closure* of the segments required to install a selected segment; consequently, under some conditions it does not properly install all the segments required to install a selected segment, and the installation process can fail in unexpected ways. During the testing of ELIST, this caused the Segment Installer to hang. Out-of-order installation can and will be tested in the future, after the Segment Installer is fixed.
- As far as was practical, all paths (involving different choices or otherwise different responses to interactions) were tested. In general, this required repeating certain steps that had already been tested to arrive quickly at a branch point from which a different path could be taken, or a different option selected.
- When the expected response was to abort the install, it was verified that the segment was left uninstalled (*i.e.*, was not included in the Segment Installer's list of installed segments).
- Successful installation was verified by checking that the segment was included in the list of installed segments following installation and that certain files and directories were created, as expected. In the test results, the evidence supporting these actions is usually presented in the form of screen captures.
- Where relevant, log files created during installation were checked to ensure that no unexpected errors were recorded there.
- Installation tests were followed by functional tests of the installed segments (after all segments were installed) to verify correct operation. Functional testing was performed by using typical user accounts (not `sysadmin`) having the requisite profiles. This serves as the ultimate test of successful installation.

### 3.3.2 Procedures for Functional Testing

Functional testing was performed by using the features of the segment being tested. The behavior of the feature was observed, and in some cases appropriate tools were used to confirm that the desired effect had been achieved.

### 3.3.3 Procedures for Deinstallation Testing

The procedures followed during deinstallation testing are similar to those for installation testing and are summarized here:

- As with installation testing, the Segment Installer was executed by the system administrator (`sysadmin` account).
- The Segment Installer was invoked from the command line in a terminal window.

- The Segment Installer was the only application running.
- Erroneous output, if any, was captured.
- As discussed in the *ELIST IP*, protection against deinstalling segments out of order is built into the deinstallation scripts. Although nothing prevents the testing of that behavior, only spot checks have been performed, and those tests are not reported here. If not deemed too difficult to express, they will be included in future versions of this document.
- When the expected response was to abort the deinstall, it was verified that the segment was left installed (*i.e.*, was still included in the Segment Installer’s list of installed segments).
- Successful deinstallation was verified by checking that the segment was no longer included in the list of installed segments following deinstallation and that certain files and directories were no longer in existence, as expected.
- Where relevant, log files created during deinstallation were checked to ensure that no unexpected errors were recorded there. Those tests are not reported here. If not deemed too difficult to express, they will be included in future versions of this document.
- Although deinstalling the ELIST Database Segment without deinstalling the ELIST Database Instance Segment does not result in the loss of database user accounts, it does cause the database user roles to be dropped from those accounts. Special processing ensures that these roles are restored when the ELIST Database Segment is reinstalled (perhaps to replace the segment with a new version). The correctness of this processing has been verified, but the tests are not reported here. If not deemed too difficult to express, they will be included in future versions of this document.

### 3.3.4 Summary of Tests Performed

The tables in this section state the goal of each test.

Table 2 summarizes the installation tests. The person performing the tests, referred to in the table as “the administrator,” must log in as `sysadmin`.

**Table 2. Summary of Installation Tests**

Segment / Test	Section
<b>ELIST Global Data Segment</b>	
Verify that the ELIST Global Data Segment installs correctly.	4.2.1.1
<b>ELIST Database Instance Segment</b>	
Verify that, if database instances already exist, the administrator is prompted to choose between <b>Use existing instance</b> and <b>Create new instance</b> .	4.2.2.1
Verify that, after choosing <b>Create new instance</b> , the administrator is prompted to enter the name to be assigned to the new instance.	4.2.2.2
Verify that the installation is aborted if the administrator clicks <b>OK</b> without entering a new instance name.	4.2.2.3
Verify that the administrator is informed, and is given another chance, if the name entered is actually that of an <i>existing</i> instance.	4.2.2.4

Verify that, after entering a new instance name, the administrator is prompted to enter a password to be assigned to the <code>SYSTEM</code> user of the new instance.	4.2.2.5
Verify that, after three unsuccessful attempts to enter and confirm the password to be assigned to the <code>SYSTEM</code> user of the new instance, the administrator is informed and the password prompt is repeated.	4.2.2.6
Verify that, after successfully entering and confirming the password to be assigned to the <code>SYSTEM</code> user of the new instance, the administrator is informed that a new database instance has been created.	4.2.2.7
Verify that the ELIST Database Instance Segment installs correctly when a new database instance is created.	4.2.2.8
Verify that, after choosing <b>Use existing instance</b> , the administrator is prompted to enter the name of an existing instance.	4.2.2.9
Verify that the installation is aborted if the administrator clicks <b>OK</b> without entering an existing instance name.	4.2.2.10
Verify that the administrator is informed, and is given another chance, if the name entered is <i>not</i> that of an existing instance.	4.2.2.11
Verify that, after entering an existing instance name, the administrator is reminded that the instance will not be removed when the segment is later deinstalled.	4.2.2.12
Verify that the ELIST Database Instance Segment installs correctly when an existing database instance is used.	4.2.2.13
Verify that, if no database instances exist, the administrator is so informed and the subsequent behavior is as if the administrator had chosen <b>Create new instance</b> .	4.2.2.14
<b>ELIST Database Fill Segment</b>	
Verify that the ELIST Database Fill Segment installs correctly.	4.2.3.1
<b>ELIST Database Segment</b>	
Verify that the administrator is prompted to enter the previously assigned password of the <code>SYSTEM</code> user of the ELIST database instance.	4.2.4.1
Verify that the installation is aborted if the administrator clicks <b>Cancel</b> instead of entering the <code>SYSTEM</code> password.	4.2.4.2
Verify that the installation is aborted if, after three attempts, the administrator fails to enter the <code>SYSTEM</code> password successfully.	4.2.4.3
Verify that, after successfully entering the <code>SYSTEM</code> password, the administrator is informed that the database has been created.	4.2.4.4
Verify that the ELIST Database Segment installs correctly.	4.2.4.5
<b>ELIST Database Utility Segment</b>	
Verify that the ELIST Database Utility Segment installs correctly.	4.2.5.1
<b>ELIST Software Segment</b>	
Verify that the ELIST Software Segment installs correctly on the database server.	4.2.6.1
Verify that the ELIST Software Segment installs correctly on a separate application client.	4.2.6.2
<b>ELIST Reference Data Segment</b>	
Verify that the ELIST Reference Data Segment installs correctly on the database server.	4.2.7.1
Verify that the ELIST Reference Data Segment installs correctly on a separate application client.	4.2.7.2

Table 3 summarizes the functional tests of the features of the ELIST Database Utility Segment. The person performing the tests, referred to in the table as “the user,” must log in as an administrative ELIST user.

**Table 3. Summary of Functional Tests of the Features of the ELIST Database Utility Segment**

Feature / Test	Section
<b>Create User Account</b>	

Verify that the user is prompted to enter the name of the database account to create.	4.3.1.1
Verify that the feature terminates if the user clicks <b>OK</b> without entering an account name.	4.3.1.2
Verify that, after entering an account name and clicking <b>OK</b> , the user is prompted to enter a password to be assigned to the new account.	4.3.1.3
Verify that the feature terminates if the user fails within three attempts to enter and confirm the password to be assigned to the new account.	4.3.1.4
Verify that, after successfully entering and confirming the password to be assigned to the new account, the user is prompted to enter the <code>SYSTEM</code> password.	4.3.1.5
Verify that the feature terminates if the user fails within three attempts to enter the <code>SYSTEM</code> password.	4.3.1.6
Verify that, after successfully entering the <code>SYSTEM</code> password, the user is informed that the database account has been created, assuming it did not initially exist.	4.3.1.7
Verify that the database account now exists.	4.3.1.8
Verify that, after successfully entering the <code>SYSTEM</code> password, the user is informed and the feature terminates if the database account already exists.	4.3.1.9
<b>Remove User Account</b>	
Verify that the user is prompted to enter the name of the database account to remove.	4.3.2.1
Verify that the feature terminates if the user clicks <b>OK</b> without entering an account name.	4.3.2.2
Verify that, after entering an account name and clicking <b>OK</b> , the user is prompted to enter the <code>SYSTEM</code> password.	4.3.2.3
Verify that the feature terminates if the user fails within three attempts to enter the <code>SYSTEM</code> password.	4.3.2.4
Verify that, after successfully entering the <code>SYSTEM</code> password, the user is informed that the database account has been removed, assuming the account initially existed.	4.3.2.5
Verify that the database account no longer exists.	4.3.2.6
Verify that, after successfully entering the <code>SYSTEM</code> password, the user is informed and the feature terminates if the database account does not exist.	4.3.2.7
<b>Lock User Account</b>	
Verify that the user is prompted to enter the name of the database account to lock.	4.3.3.1
Verify that the feature terminates if the user clicks <b>OK</b> without entering an account name.	4.3.3.2
Verify that, after entering an account name and clicking <b>OK</b> , the user is prompted to enter the <code>SYSTEM</code> password.	4.3.3.3
Verify that the feature terminates if the user fails within three attempts to enter the <code>SYSTEM</code> password.	4.3.3.4
Verify that, after successfully entering the <code>SYSTEM</code> password, the user is informed that the database account has been locked, assuming the account exists.	4.3.3.5
Verify that the database account has been locked.	4.3.3.6
Verify that, after successfully entering the <code>SYSTEM</code> password, the user is informed and the feature terminates if the database account does not exist.	4.3.3.7
<b>Unlock User Account</b>	
Verify that the user is prompted to enter the name of the database account to unlock.	4.3.4.1
Verify that the feature terminates if the user clicks <b>OK</b> without entering an account name.	4.3.4.2
Verify that, after entering an account name and clicking <b>OK</b> , the user is prompted to enter the <code>SYSTEM</code> password.	4.3.4.3
Verify that the feature terminates if the user fails within three attempts to enter the <code>SYSTEM</code> password.	4.3.4.4
Verify that, after successfully entering the <code>SYSTEM</code> password, the user is informed that the database account has been unlocked, assuming the account exists.	4.3.4.5
Verify that the database account has been unlocked.	4.3.4.6

Verify that, after successfully entering the <i>SYSTEM</i> password, the user is informed and the feature terminates if the database account does not exist.	4.3.4.7
<b>List User Accounts</b>	
Verify that the user is prompted to enter the <i>SYSTEM</i> password.	4.3.5.1
Verify that the feature terminates if the user fails within three attempts to enter the <i>SYSTEM</i> password.	4.3.5.2
Verify that, after successfully entering the <i>SYSTEM</i> password, the user is presented with a list of existing accounts.	4.3.5.3
<b>Change User Password</b>	
Verify that the user is prompted to enter the name of the database account whose password is to be changed.	4.3.6.1
Verify that the feature terminates if the user clicks <b>OK</b> without entering an account name.	4.3.6.2
Verify that, after entering an account name and clicking <b>OK</b> , the user is prompted to enter a new password to be assigned to the account.	4.3.6.3
Verify that the feature terminates if the user fails within three attempts to enter and confirm the new password to be assigned to the account.	4.3.6.4
Verify that, after successfully entering and confirming the new password to be assigned to the account, the user is prompted to enter the <i>SYSTEM</i> password.	4.3.6.5
Verify that the feature terminates if the user fails within three attempts to enter the <i>SYSTEM</i> password.	4.3.6.6
Verify that, after successfully entering the <i>SYSTEM</i> password, the user is informed that the password of the database account has been changed, assuming the account exists.	4.3.6.7
Verify that the password of the database account has been changed.	4.3.6.8
Verify that, after successfully entering the <i>SYSTEM</i> password, the user is informed and the feature terminates if the database account does not exist.	4.3.6.9

Table 4 summarizes the functional tests of the administrative features of the ELIST Software Segment. The person performing the tests, referred to in the table as “the user,” must log in as an administrative ELIST user.

**Table 4. Summary of Functional Tests of the Administrative Features of the ELIST Software Segment**

<b>Feature / Test</b>	<b>Section</b>
<b>Add Map Data feature</b>	
Verify that the user is prompted to insert a CD and then choose between <b>Proceed</b> and <b>Cancel</b> .	4.4.1.1
Verify that the prompt is repeated if the user clicks <b>Proceed</b> before inserting a CD.	4.4.1.2
Verify that the feature terminates if <b>Cancel</b> is clicked.	4.4.1.3
Verify that the prompt is repeated if the user clicks <b>Proceed</b> after inserting a CD but before the File Manager window for the CD opens.	4.4.1.4
Verify that the prompt is <i>not</i> repeated if the user clicks <b>Proceed</b> after the File Manager window for the CD opens.	4.4.1.5
Verify that, if the map data set on the CD is not already present in the ELIST Reference Data Segment, it is added if space is available.	4.4.1.6
Verify that, if the map data set on the CD <i>is</i> already present in the ELIST Reference Data Segment, it is <i>not</i> added.	4.4.1.7
Verify that, if the map data set on the CD is not already present in the ELIST Reference Data Segment, it is <i>not</i> added if space is unavailable.	4.4.1.8
Verify that the feature works correctly on a separate application client.	4.4.1.9

Delete Map Data feature	
Verify that, if one or more map data sets are present in the ELIST Reference Data Segment, the user is presented with a list of the their names and is prompted to enter one of them.	4.4.2.1
Verify that the feature terminates if the user clicks <b>OK</b> without entering the name of a map data set.	4.4.2.2
Verify that, if the name entered is <i>not</i> actually that of an existing map data set, the user is informed and is given another chance.	4.4.2.3
Verify that, if the user enters the name of an existing map data set, the map data set is deleted.	4.4.2.4
Verify that, if no map data sets are present in the ELIST Reference Data Segment, the user is so informed.	4.4.2.5
Verify that the feature works correctly on a separate application client.	4.4.2.6

Table 5 summarizes the functional tests of the general features of the ELIST Software Segment. The person performing the tests, referred to in the table as “the user,” must log in as a general ELIST user.

**Table 5. Summary of Functional Tests of the General Features of the ELIST Software Segment**

Feature / Test	Section
<b>Run ELIST feature</b>	
Verify that general ELIST users can launch ELIST.	4.5.1.1
Verify that the feature can be terminated.	4.5.1.2
Verify that the feature works correctly on a separate application client.	4.5.1.3
<b>Run ETEdit feature</b>	
Verify that general ELIST users can launch ETEdit.	4.5.2.1
Verify that the feature can be terminated.	4.5.2.2
Verify that the feature works correctly on a separate application client.	4.5.2.3

**NOTE:** Obviously, many capabilities of the two features listed above are not tested in the current version of this document. Additional tests will be designed and incorporated in the future.

Table 6 summarizes the deinstallation tests. The person performing the tests, referred to in the table as “the administrator,” must log in as `sysadmin`.

**Table 6. Summary of Deinstallation Tests**

Segment / Test	Section
<b>ELIST Reference Data Segment</b>	
Verify that the ELIST Reference Data Segment deinstalls correctly on the database server.	4.6.1.1
Verify that the ELIST Reference Data Segment deinstalls correctly on a separate application client.	4.6.1.2
<b>ELIST Software Segment</b>	
Verify that the ELIST Software Segment deinstalls correctly on the database server.	4.6.2.1
Verify that the ELIST Software Segment deinstalls correctly on a separate application client.	4.6.2.2
<b>ELIST Database Utility Segment</b>	
Verify that the ELIST Database Utility Segment deinstalls correctly.	4.6.3.1
<b>ELIST Database Segment</b>	
Verify that the administrator is prompted to enter the previously assigned password of the <code>SYSTEM</code> user of the ELIST database instance.	4.6.4.1

Verify that the deinstallation is aborted if the administrator clicks <b>Cancel</b> instead of entering the <code>SYSTEM</code> password.	4.6.4.2
Verify that the deinstallation is aborted if, after three attempts, the administrator fails to enter the <code>SYSTEM</code> password successfully.	4.6.4.3
Verify that, after successfully entering the <code>SYSTEM</code> password, the administrator is informed that the database has been removed.	4.6.4.4
Verify that the ELIST Database Segment deinstalls correctly.	4.6.4.5
<b>ELIST Database Fill Segment</b>	
Verify that the ELIST Database Fill Segment deinstalls correctly.	4.6.5.1
<b>ELIST Database Instance Segment</b>	
Verify that, if <b>Create new instance</b> was chosen when the segment was installed, the administrator is informed that the database instance has been removed.	4.6.6.1
Verify that the ELIST Database Instance Segment deinstalls correctly when its installation created a new database instance.	4.6.6.2
Verify that, if <b>Use existing instance</b> was chosen when the segment was installed, the administrator is informed that the database instance has <i>not</i> been removed.	4.6.6.3
Verify that the ELIST Database Instance Segment deinstalls correctly when its installation used an existing database instance..	4.6.6.4
<b>ELIST Global Data Segment</b>	
Verify that the ELIST Global Data Segment deinstalls correctly.	4.6.7.1

**NOTE:** Special logic has been implemented in the installation and deinstallation scripts of the ELIST Database Segment to record the names of the existing ELIST database user accounts when the the ELIST Database Segment is deinstalled and to use those names when the segment is later reinstalled (as, for example, when the segment is being replaced with a later version) to recreate the accounts. However, the current version of this document does not include tests of that logic. They will be included in a later version of this document.

### 3.4 Test Environment

The test site at ANL is the DII COE development and testing laboratory in the Decision and Information Sciences Division. At the time of testing, the laboratory included two Sun Unix workstations running the Solaris operating system; it also had a PC running Windows NT reserved for future use. One of the Sun workstations is configured as a database server, while the other is an application client. (The application client segments, Segments 6 and 7 in Table 1, are capable of being installed on either an application client or a database server, and have been so tested; in the latter case, the database server machine is said to be configured both as a database server and an application client.)

Both Sun workstations are on the local LAN. They communicate freely with each other and share certain file systems, using NFS. Both have outward access (by `ftp` and `rlogin`) to selected other machines on the network, but no other machine is able to access the DII COE testbed machines. Except for the file systems shared with each other, no file system sharing takes place with the rest of the network.

Care was used during testing to ensure that no artifacts of the development environment affected the results of the testing.

#### 3.4.1 Hardware

The database server is a Sun Ultra 1 (named `bologna`) with 256 MB of RAM installed, while the separate application client is a Sun SPARC 20 (named `delft`) with 96 MB of RAM

installed.<sup>4</sup> Because ELIST requires large amounts of disk space, especially on the database server, both machines have additional hard disks installed. When the Solaris operating system was installed, the disk partitions were given the custom sizes shown in Table 7.

**Table 7. Size of Disk Partitions on the Testbed Machines**

Partition	Size (MB)	
	Database Server	Application Client
/	500	800
/var	300	200
/tmp	600	900
/security1	100	100
/security2	100	100
/ora01	3000	701
/ora02	3000	N/A
/ora03	3000	N/A
/ora04	3000	N/A
/h	726	6958
/home1	1014	N/A
/home2	1013	N/A
/home3	2633	N/A

Note that the Oracle partitions on the database server are larger than the “feasible minimum” sizes documented in the *ELIST IP*.

On both machines, the /tmp, or swap, partition is divided between two physical drives for performance reasons.

Though not reflected in Table 7, the /h/data/global and /h/USERS/global partitions, or mount points, are shared by the database client and the application client server, using NFS; they are exported by the server and imported by the client. This sharing is required if global, or “roaming,” users are to be supported, and it is also required to support data segments of global scope, such as the ELIST Global Data Segment. The ORACLE RDBMS (ORAS) and ORACLE DataBase Instance (ORADBI) segments also use it to share data globally, as documented in the *Installation Procedures (IP) for the ORACLE Client Applications (ORAC)* segment.

### 3.4.2 Software

Solaris 7 is installed on both machines.

Both machines have the DII COE Kernel, version 4.4 (*aka* 4.2.0.0P4), and the Solaris Patch Update (SOLPTH) segment, version 4.2.0.0, installed.

The infrastructure service segments installed on each machine to support ELIST are as shown in Table 8. Additional COE-component, infrastructure service, and common application segments are also installed, but they are not specifically required for ELIST and are therefore omitted from the table.

<sup>4</sup> A Sun SPARC 20 is generally too small and too slow to be an effective machine on which to host ELIST, but it was marginally adequate for testing.

**Table 8. Infrastructure Service Segments Required by ELIST**

Segment			Database Server	Application Client
Name	Prefix	Version		
Java Platform 1	JAVA1	4.0.1.0/1.1.8_12	X	
Java Platform 2	JAVA2	4.3.0.0	X	X
Database Administrator Server	DBAdmS	3.0.2.0	X	
Database Administrator Runtime	DBAdmR	3.0.2.0	X	
ORACLE RDBMS	ORAS	2.1.0.0/8.1.6.1	X	
ORACLE DataBase Instance	ORADBI	2.1.0.0/8.1.6	X	
ORACLE DataBase Administration	ORADBA	2.1.0.0/2.1.0		X
ORACLE Client Applications	ORAC	2.1.0.0/8.1.6		X

### 3.4.3 Operating System Kernel Parameters

The ORACLE RDBMS (ORAS) segment requires certain minimum values for the operating system kernel parameters governing shared memory and semaphore use (see the *ELIST IP*). The values of these parameters defined in the `/etc/system` file on the database server are as shown in Table 9.

**Table 9. Solaris Kernel Parameter Values on the Database Server**

Parameter	Value
<b>shmmx</b>	4294967295
<b>shmmn</b>	1
<b>shmmni</b>	1024
<b>shmseg</b>	1024
<b>semmsl</b>	400
<b>semnmi</b>	120
<b>semnms</b>	200
<b>semnmu</b>	400

This page intentionally left blank.

## 4. Test Descriptions and Results

This section describes each test in detail, giving the steps involved in performing the test, the expected results, and the actual (or observed) results.

### 4.1 Conventions

#### 4.1.1 Prerequisite Conditions

The hardware and software prerequisites for performing the tests described in this section are documented in Section 0. This document assumes that none of the ELIST segments are installed before testing begins.

#### 4.1.2 Test Inputs and Procedures

Each test described in this section contains all necessary information concerning the test inputs and procedures.

#### 4.1.3 Expected Test Results and Criteria for Evaluating Test Results

Each test in described in this section includes the results expected from performing the test. The actual results of performing the test are also documented. The criterion for evaluating the actual test results is agreement with the expected results.

#### 4.1.4 Order of the Tests

The tests are presented in a sequential order. If a test does not include specific mention of its successor, go on to the next section when finishing a test. In order to test alternative choices, where they are presented in the installation procedures, you will occasionally be directed to back up and reestablish a previous state, then take a different path out of that state. Always follow the directions about what test to perform next, if present.

## 4.2 Installation Tests and Results

This tests in this section demonstrate that all the segments of the ELIST mission application install correctly. In these tests, the segments are installed in the proper order. (Section 3.3.1 explains that a bug in the current version of the Segment Installer unintentionally allows it to install segments out of order under certain conditions, making it not currently feasible to verify that the segment dependencies have been accurately stated in segment descriptors.)

Log in to a database server platform as the system administrator (`sysadmin` user). Follow the instructions in this section for installing the segments. The instructions here make liberal reference to the procedures for installing particular segments as presented in the *ELIST IP*; consult the *ELIST IP* when directed to do so.

### 4.2.1 ELIST Global Data Segment

Because it is not interactive, demonstrating that the installation of the ELIST Global Data Segment is performed correctly requires only one test.

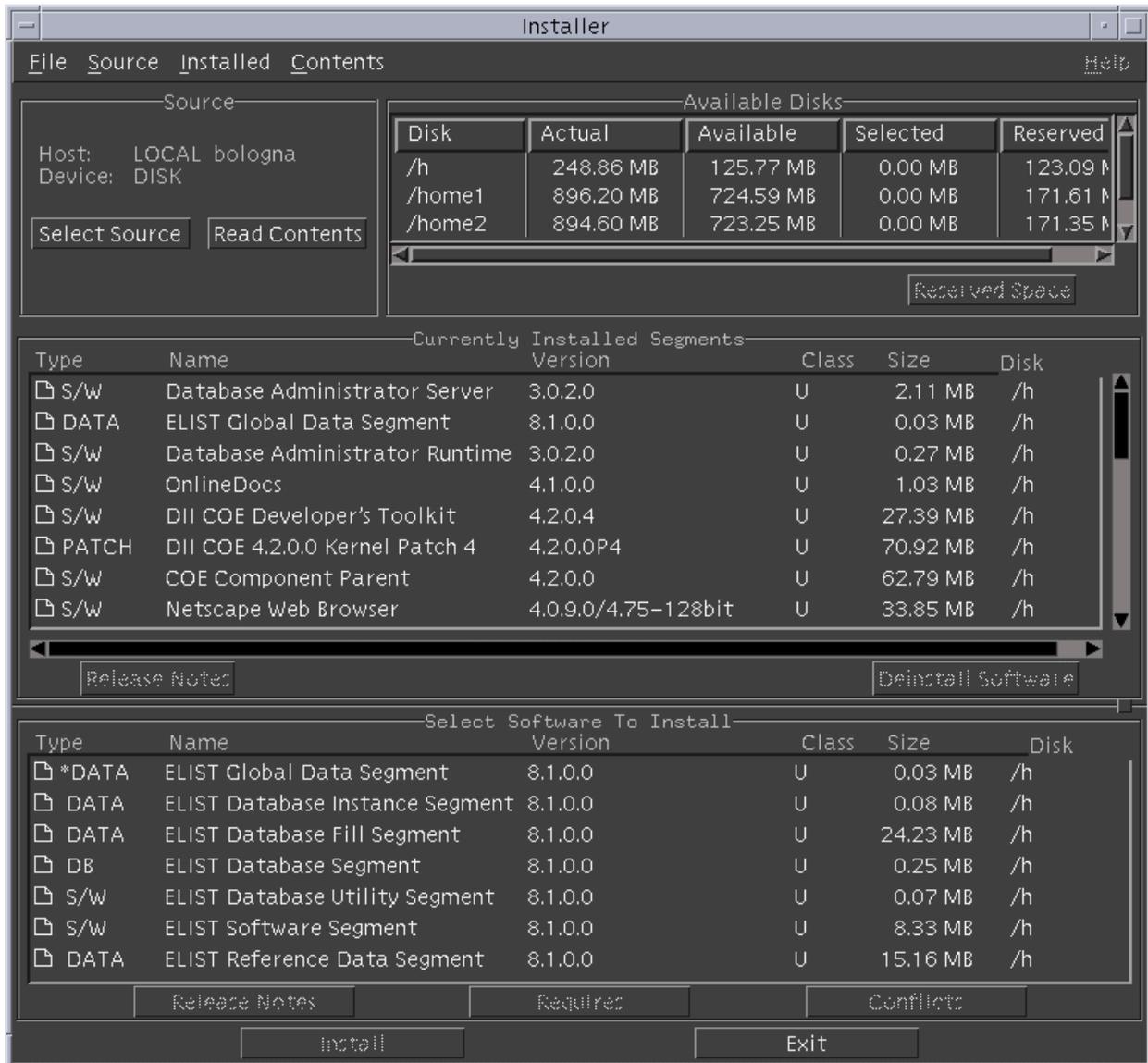
#### 4.2.1.1 Test IGDS-FinishInstall

Goal: Verify that the ELIST Global Data Segment installs correctly.

Test Description: Follow the steps in the *ELIST IP* for installing the ELIST Global Data Segment.

Expected Results: The Segment Installer should complete the installation of the ELIST Global Data Segment.

Test Results: As shown in Figure 2, the Segment Installer window reopens at the end of the installation process, with the ELIST Global Data Segment in the list of currently installed segments.



**Figure 2. Segment Installer Window after Installing the ELIST Global Data Segment**

Test Status: Passed.

## 4.2.2 ELIST Database Instance Segment

Because it is interactive, demonstrating that the installation of the ELIST Database Instance Segment is performed correctly requires a sequence of small tests.

### 4.2.2.1 Test IDIS-PromptForChoice

Goal: Verify that, if database instances already exist, the administrator is prompted to choose between **Use existing instance** and **Create new instance**.

Test Description: This test can only be performed if one or more database instances already exist.

**NOTE:** If database instances do not already exist and you do not wish to create them now, proceed to Section 4.2.2.14 instead to perform Test IDIS-PromptForNewInstanceName2. The latter tests cover what is expected to be the normal situation and are far easier to perform.

If no database instances exist, consult the section titled “Creating an ELIST Database Instance Prior to Installing the ELIST Database Instance Segment” in the *System Administrator’s Manual (SAM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Instance Segment* and then perform the following steps:

- Log in as a DBA (see the “Account Preparation” section of the *ELIST IP*).
- Use the **Create Instances** feature of the **ORADBI** application, following the instructions in the *System Administrator’s Manual (SAM) for ORACLE Client Applications (ORAC), ORACLE DataBase Administration (ORADBA), ORACLE DataBase Instance (ORADBI), and ORACLE RDBMS (ORAS)*, to create a database instance of medium or large size.
- Use the **SQL Plus** feature of the **ORAS** application to enter the following SQL commands, which create a large rollback segment with a non-standard name that ELIST requires. (This step may be skipped if you do *not* actually intend to create a database in the instance you have just created and ultimately use ELIST with that database.)

```
CREATE TABLESPACE BIGROLL
    DATAFILE
        '/ora03/oradata/<ORACLE_SID>/bigroll.dbf'
    SIZE 500M
    REUSE
    AUTOEXTEND ON NEXT 10M
    MINIMUM EXTENT 10M
    DEFAULT STORAGE
        (INITIAL 10M
        NEXT 10M
        MINEXTENTS 2
        MAXEXTENTS 4096);

CREATE PUBLIC ROLLBACK SEGMENT BIG_RBS
    TABLESPACE BIGROLL
    STORAGE (OPTIMAL 40M);
```

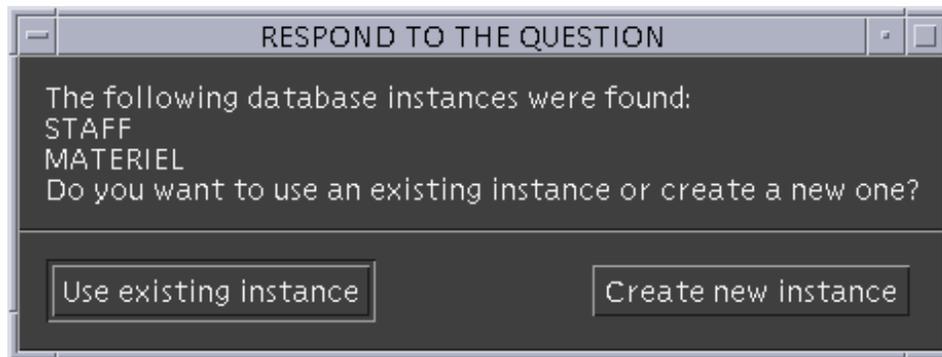
```
ALTER ROLLBACK SEGMENT "BIG_RBS" ONLINE;
```

In the preceding SQL commands, <ORACLE\_SID> is the name of the instance you created in the previous step.

- Log out, log back in as `sysadmin`, and follow the steps in the *ELIST IP* for installing the ELIST Database Instance Segment, stopping when first prompted to make a choice.

Expected Results: The Segment Installer should display the names of existing instances and prompt the administrator to make a choice between using one of them and creating a new instance.

Test Results: The window shown in Figure 3 opens. (The existing database instances were named **STAFF** and **MATERIEL**.)



**Figure 3. Window for Expressing the Principal Choice While Installing the ELIST Database Instance Segment**

Test Status: Passed.

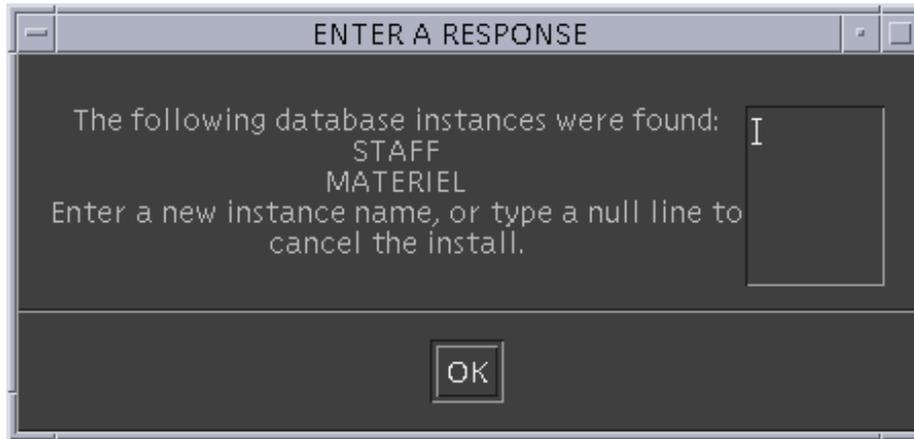
#### 4.2.2.2 Test IDIS-PromptForNewInstanceName1

Goal: Verify that, after choosing **Create new instance**, the administrator is prompted to enter the name to be assigned to the new instance.

Test Description: In the currently open window, which is shown in Figure 3, click **Create new instance**. (The alternate choice is tested in Test IDIS-PromptForExistingInstanceName in Section 4.2.2.9.)

Expected Results: The Segment Installer should display the names of the existing instances again and prompt the administrator to enter the name to be assigned to the new instance (a name *not* in the list).

Test Results: The window shown in Figure 4 opens.



**Figure 4. Window for Entering a New Instance Name When Instances Already Exist**

Test Status: Passed.

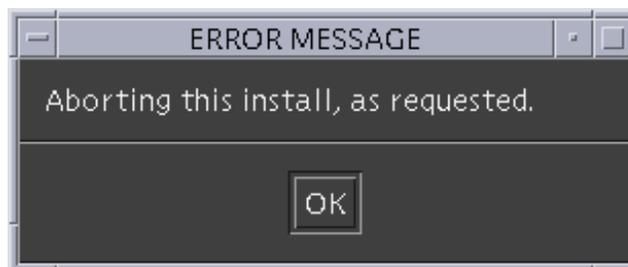
#### 4.2.2.3 Test IDIS-LeaveNewInstanceNameEmpty

Goal: Verify that the installation is aborted if the administrator clicks **OK** without entering a new instance name.

Test Description: In the currently open window, which is shown in Figure 4 or Figure 16, leave the textbox empty and click **OK**.

Expected Results: The Segment Installer should abort the installation of the ELIST Database Instance Segment.

Test Results: The window shown in Figure 5 opens.



**Figure 5. Window Following Failure to Enter a New Instance Name**

Upon clicking **OK**, the Segment Installer window shown in Figure 2 reopens. (The ELIST Database Instance Segment is not in the list of currently installed segments.)

Test Status: Passed.

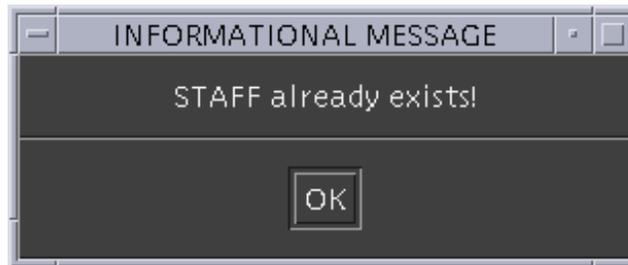
#### 4.2.2.4 Test IDIS-EnterErroneousNewInstanceName

Goal: Verify that the administrator is informed, and is given another chance, if the name entered is actually that of an *existing* instance.

Test Description: Repeat Test IDIS-PromptForChoice in Section 4.2.2.1 and Test IDIS-PromptForNewInstanceName1 in Section 4.2.2.2. In the window that opens as a result, which is shown in Figure 4 or Figure 16, enter the name of one of the *existing* instances (say, **STAFF**).

Expected Results: The Segment Installer should inform the administrator that the instance exists and should repeat the prompt for a new instance name.

Test Results: The window shown in Figure 6 opens.



**Figure 6. Message Informing that Database Instance Exists**

Upon clicking **OK**, the window shown in Figure 4 reopens.

Test Status: Passed.

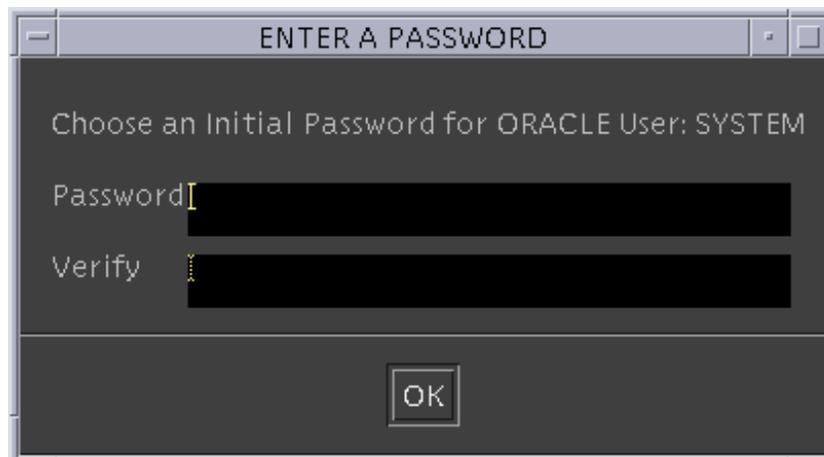
#### 4.2.2.5 Test IDIS-PromptForSystemPassword

Goal: Verify that, after entering a new instance name, the administrator is prompted to enter a password to be assigned to the `SYSTEM` user of the new instance.

Test Description: In the currently open window, which is shown in Figure 4 or Figure 16, enter into the textbox a name *not* in the list of existing instances (say, **ELIST1**) and click **OK**.

Expected Results: The Segment Installer should prompt the administrator to enter a password to be assigned to the `SYSTEM` user of the new instance.

Test Results: The window shown in Figure 7 opens.



**Figure 7. Window Prompting for Password to be Assigned to `SYSTEM` User of the New Database Instance**

Test Status: Passed.

#### 4.2.2.6 Test IDIS-FailToAssignSystemPassword

Goal: Verify that, after three unsuccessful attempts to enter and confirm the password to be assigned to the `SYSTEM` user of the new instance, the administrator is informed and the password prompt is repeated.

Test Description: In the currently open window, which is shown in Figure 7, enter, three times in succession, information *other than* the same valid password in the **Password** field and the **Verify** field, clicking **OK** after doing so each time. For example, you may:

- leave both fields, or just the **Verify** field, empty;
- enter different passwords in both fields;
- enter the same short password (less than 6 characters) in both fields; or
- enter the same long password (more than 14 characters) in both fields.

In each case, a message window opens to explain your error. (This behavior is programmed into the `COEPromptPassword` tool, not into the `ELIST Database Instance Segment`.) After the first two failures, click **OK** to close that window and reopen the window shown in Figure 7. After the third failure, click **OK** again.

Expected Results: The Segment Installer should inform the administrator that a password has not been successfully assigned to the `SYSTEM` user of the new instance and should repeat the prompt for the password.

Test Results: The window shown in Figure 8 opens.



**Figure 8. Message Noting Failure to Assign `SYSTEM` Password**

Upon clicking **OK**, the window shown in Figure 7 reopens.

Test Status: Passed.

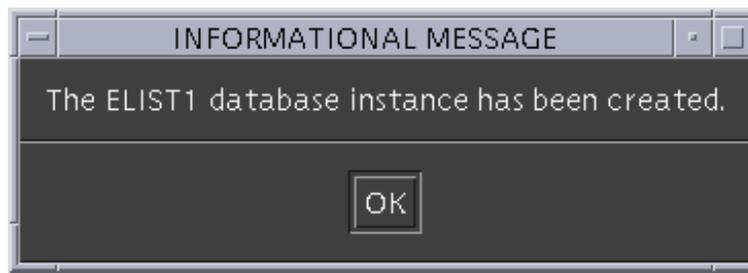
#### 4.2.2.7 Test IDIS-CreateNewInstance

Goal: Verify that, after successfully entering and confirming the password to be assigned to the SYSTEM user of the new instance, the administrator is informed that a new database instance has been created.

Test Description: In the currently open window, which is shown in Figure 7, enter the same valid password in the **Password** field and the **Verify** field (*i.e.*, a password between 6 and 14 characters in length) and click **OK**.

Expected Results: After a long delay, the Segment Installer should inform the administrator that the new database instance has been successfully created.

Test Results: The window shown in Figure 9 opens.



**Figure 9. Message Noting Successful Creation of New Database Instance**

Test Status: Passed.

#### 4.2.2.8 Test IDIS-FinishInstall

Goal: Verify that the ELIST Database Instance Segment installs correctly when a new database instance is created.

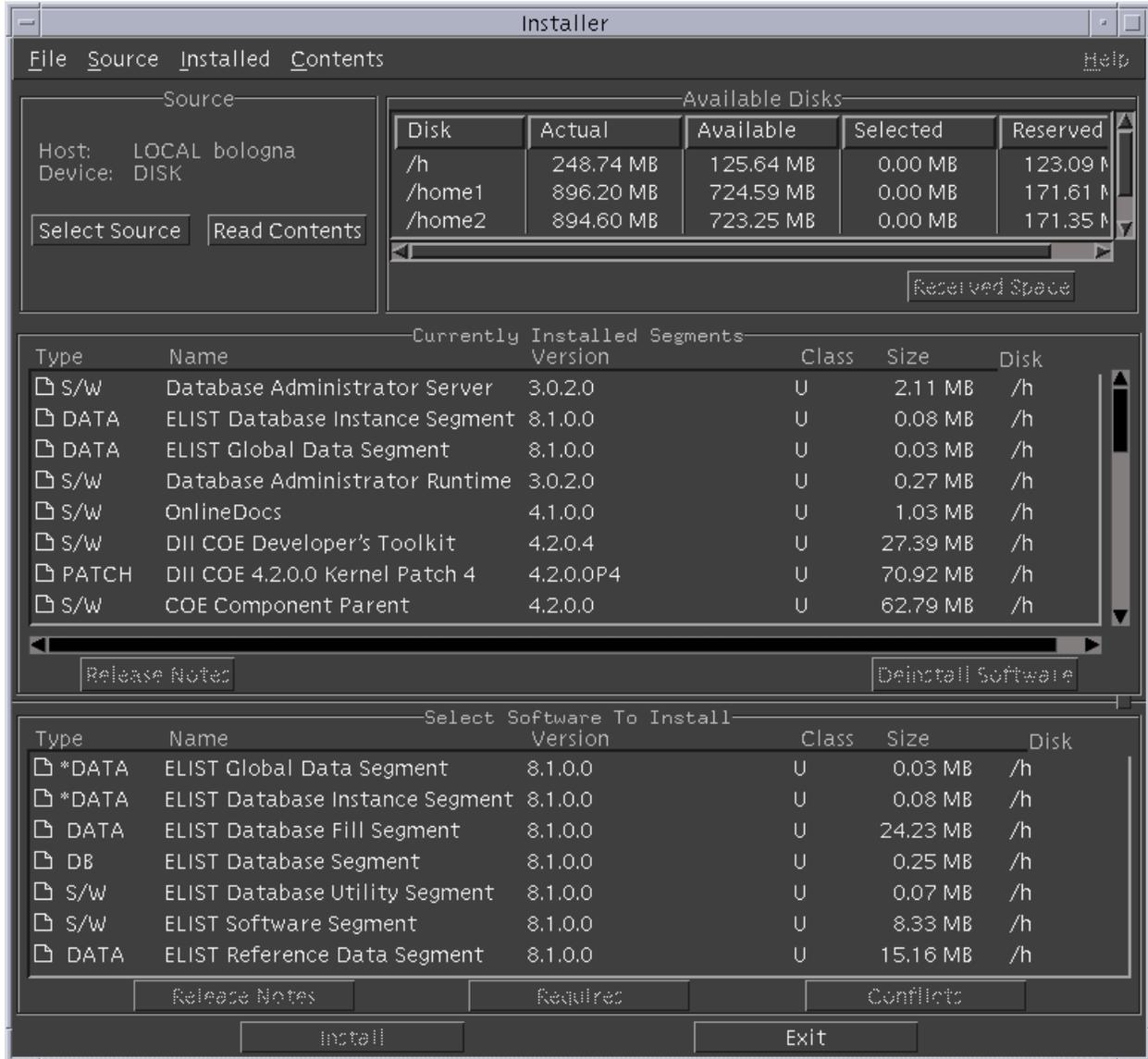
Test Description: In the currently open window, which is shown in Figure 9, click **OK**.

Expected Results: The Segment Installer should complete the installation of the ELIST Database Instance Segment, and the following five directories and one file should exist:

- /ora01/oradata/ELIST1
- /ora02/oradata/ELIST1
- /ora03/oradata/ELIST1
- /ora04/oradata/ELIST1
- /ora01/app/orace/admin/ELIST1

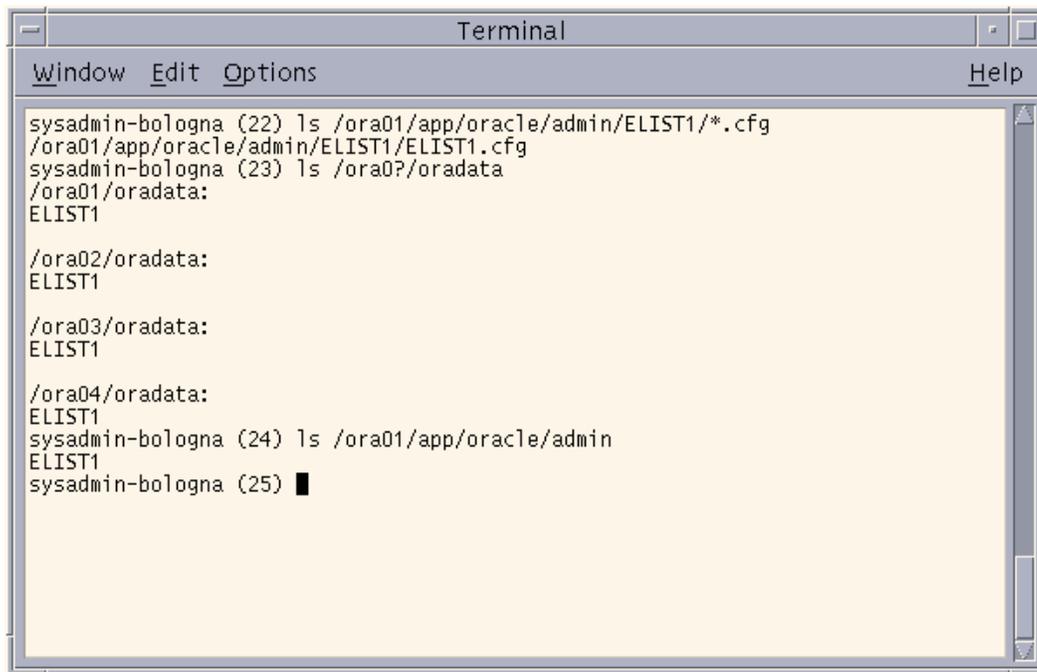
- /ora01/app/oracle/admin/ELIST1/ELIST1.cfg

**Test Results:** As shown in Figure 10, the Segment Installer window reopens, with the ELIST Database Instance Segment in the list of currently installed segments.



**Figure 10. Segment Installer Window after Installing the ELIST Database Instance Segment**

Additionally, as shown in Figure 11, the directories and file named above exist.



```

Terminal
Window Edit Options Help
sysadmin-bologna (22) ls /ora01/app/oracle/admin/ELIST1/*.cfg
/ora01/app/oracle/admin/ELIST1/ELIST1.cfg
sysadmin-bologna (23) ls /ora01/oradata
/ora01/oradata:
ELIST1

/ora02/oradata:
ELIST1

/ora03/oradata:
ELIST1

/ora04/oradata:
ELIST1
sysadmin-bologna (24) ls /ora01/app/oracle/admin
ELIST1
sysadmin-bologna (25) █

```

**Figure 11. Terminal Window Showing that Instance Directories and Files Exist**

Test Status: Passed.

Next Test: If this test was preceded by Test IDIS-CreateNewInstance in Section 4.2.2.7 and if Test IDIS-PromptForNewInstanceName2 in Section 4.2.2.14 has not yet been performed, proceed to Test IDIS-PromptForExistingInstanceName in the next section; if Test IDIS-PromptForNewInstanceName2 *has* already been performed, proceed to Test IDFS-FinishInstall in Section 4.2.3.1.

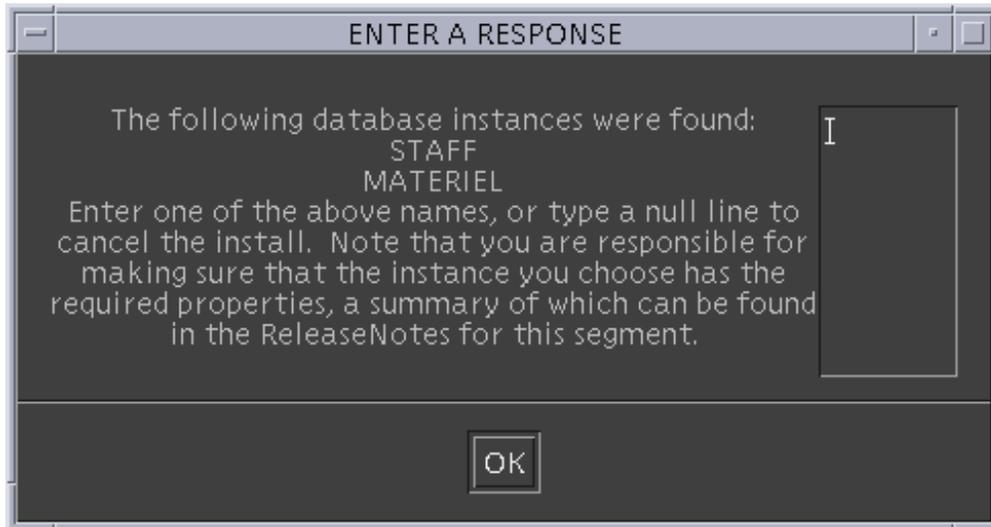
#### 4.2.2.9 Test IDIS-PromptForExistingInstanceName

Goal: Verify that, after choosing **Use existing instance**, the administrator is prompted to enter the name of an existing instance.

Test Description: Follow the instructions in the *ELIST IP* for *deinstalling* the ELIST Database Instance Segment. Then repeat Test IDIS-PromptForChoice in Section 4.2.2.1. In the window that opens as a result, which is shown in Figure 3, click **Use existing instance**.

Expected Results: The Segment Installer should display the names of the existing instances and prompt the administrator to enter one of them.

Test Results: The window shown in Figure 12 opens.



**Figure 12. Window for Entering an Existing Instance Name**

Test Status: Passed.

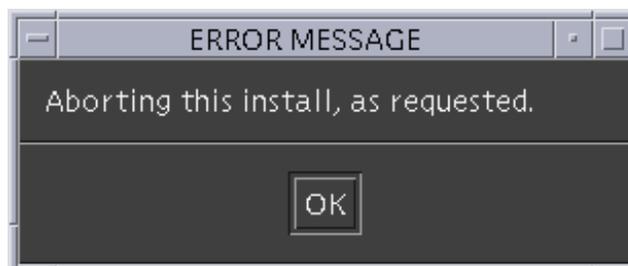
#### **4.2.2.10 Test IDIS-LeaveExistingInstanceNameEmpty**

Goal: Verify that the installation is aborted if the administrator clicks **OK** without entering an existing instance name.

Test Description: In the currently open window, which is shown in Figure 12, leave the textbox empty and click **OK**.

Expected Results: The Segment Installer should abort the installation of the ELIST Database Instance Segment.

Test Results: The window shown in Figure 13 opens.



**Figure 13. Window Following Failure to Enter an Existing Instance Name**

Upon clicking **OK**, the Segment Installer window shown in Figure 2 reopens. (The ELIST Database Instance Segment is absent from the list of currently installed segments.)

Test Status: Passed.

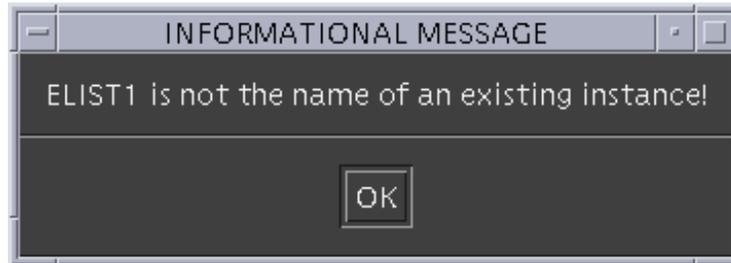
#### **4.2.2.11 Test IDIS-EnterErroneousExistingInstanceName**

Goal: Verify that the administrator is informed, and is given another chance, if name entered is *not* that of an existing instance.

Test Description: Repeat Test IDIS-PromptForExistingInstanceName in Section 4.2.2.9 and Test IDIS-LeaveExistingInstanceNameEmpty in Section 4.2.2.10. In the window that opens as a result, which is shown in Figure 12, enter the name of a non-existent instance (say, **ELIST1**).

Expected Results: The Segment Installer should inform the administrator that the instance does not exist and should repeat the prompt for an existing instance name.

Test Results: The window shown in Figure 14 opens.



**Figure 14. Message Informing that Database Instance Does not Exist**

Upon clicking **OK**, the window shown in Figure 12 reopens.

Test Status: Passed.

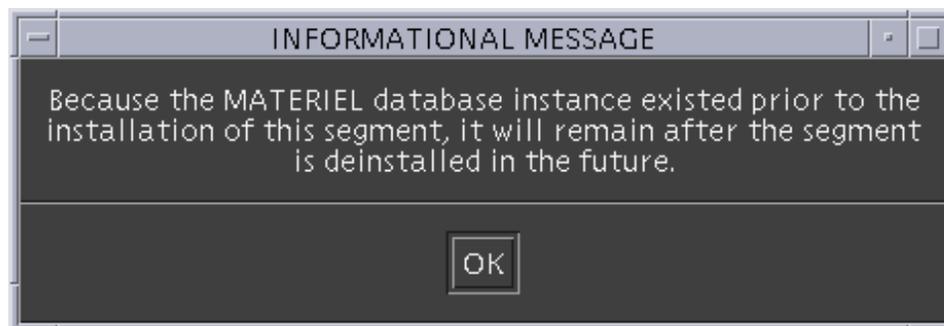
#### **4.2.2.12 Test IDIS-RemindInstanceExists**

Goal: Verify that, after entering an existing instance name, the administrator is reminded that the instance will not be removed when the segment is later deinstalled.

Test Description: In the currently open window, which is shown in Figure 12, enter the name of an existing instance (say, **MATERIEL**) and click **OK**.

Expected Results: The Segment Installer should remind the administrator that the existing instance will not be removed when the segment is later deinstalled.

Test Results: The window shown in Figure 15 opens.



**Figure 15. Window Reminding that Existing Instance Will Persist after Future Segment Deinstallation**

Test Status: Passed.

#### 4.2.2.13 Test IDIS-FinishInstall2

Goal: Verify that the ELIST Database Instance Segment installs correctly when an existing database instance is used.

Test Description: In the currently open window, which is shown in Figure 15, click **OK**.

Expected Results: The Segment Installer should complete the installation of the ELIST Database Instance Segment.

Test Results: As shown in Figure 10, the Segment Installer window reopens, with the ELIST Database Instance Segment in the list of currently installed segments.

Test Status: Passed.

#### 4.2.2.14 Test IDIS-PromptForNewInstanceName2

Goal: Verify that, if no database instances exist, the administrator is so informed and the subsequent behavior is as if the administrator had chosen **Create new instance**.

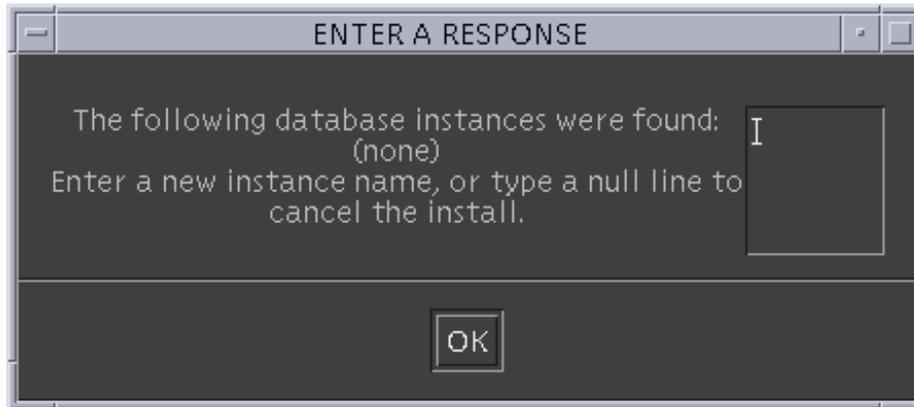
Test Description: This test can only be performed if no database instances exist.

**NOTE:** If database instances already exist and you do not wish to delete them now, proceed to Section 4.2.3.1 to perform Test IDFS-FinishInstall.

If database instances exist, first remove each existing instance, either by *deinstalling* the ELIST Database Instance Segment (in the case that the instance was created by the installation of that segment) or by logging in as a DBA (see the “Account Preparation” section of the *ELIST IP*) and deleting the database instance by using the **Delete Instances** feature of the **ORADBI** application, following the instructions in the *System Administrator’s Manual (SAM) for ORACLE Client Applications (ORAC), ORACLE DataBase Administration (ORADBA), ORACLE DataBase Instance (ORADBI), and ORACLE RDBMS (ORAS)*. Then log out, log back in as `sysadmin`, and follow the steps in the *ELIST IP* for installing the ELIST Database Instance Segment, stopping when prompted to enter a new instance name.

Expected Results: The Segment Installer should inform the administrator that no instances exist and prompt the administrator to enter the name of a new database instance.

Test Results: The window shown in Figure 16 opens.



**Figure 16. Window for Entering a New Instance Name  
When No Instances Exist**

Test Status: Passed.

Next Test: Proceed to Test IDIS-LeaveNewInstanceNameEmpty in Section 4.2.2.3.

### **4.2.3 ELIST Database Fill Segment**

Because it is not interactive, demonstrating that the installation of the ELIST Database Fill Segment is performed correctly requires only one test.

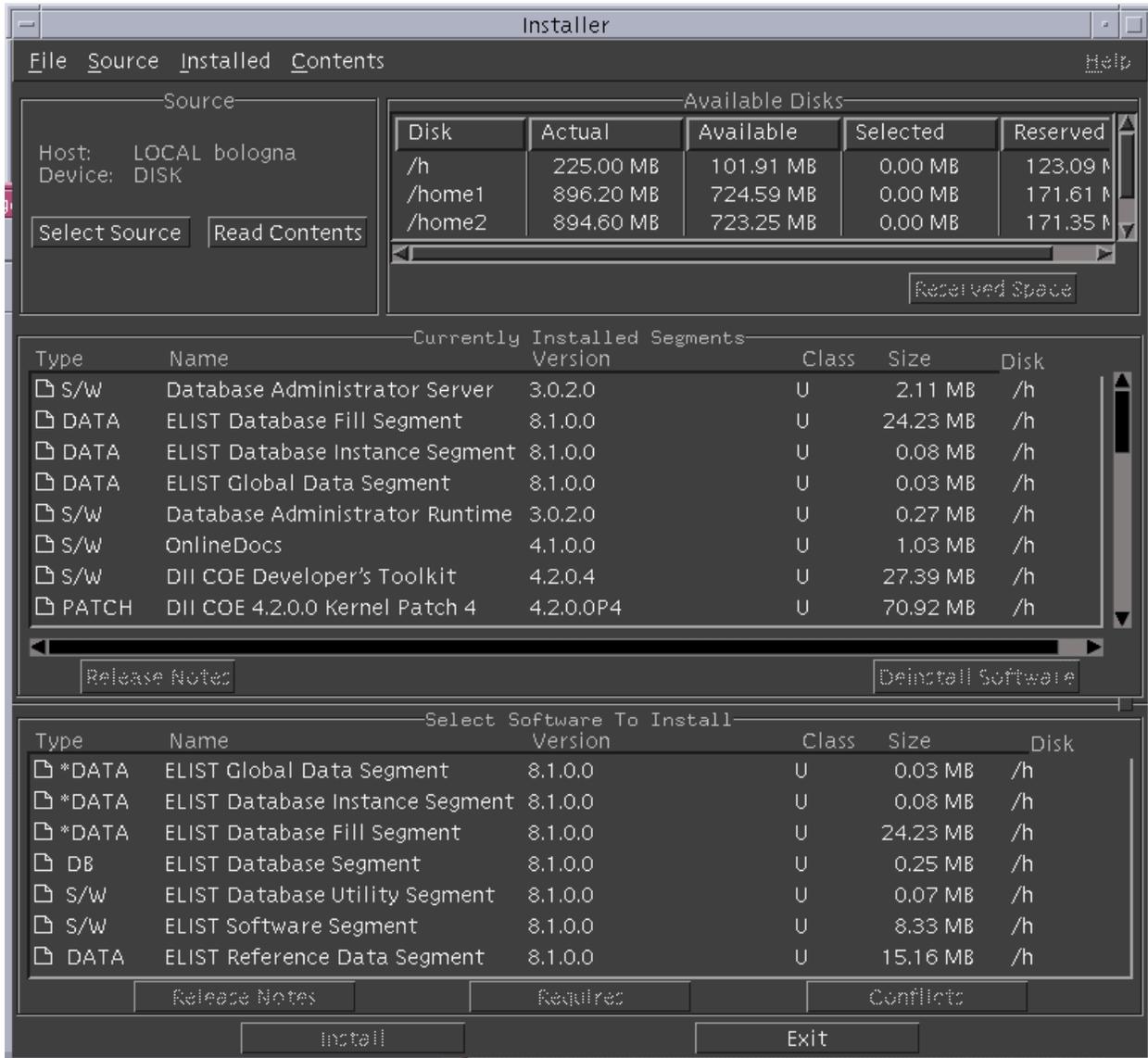
#### **4.2.3.1 Test IDFS-FinishInstall**

Goal: Verify that the ELIST Database Fill Segment installs correctly.

Test Description: Follow the steps in the *ELIST IP* for installing the ELIST Database Fill Segment.

Expected Results: The Segment Installer should complete the installation of the ELIST Database Fill Segment.

Test Results: As shown in Figure 17, the Segment Installer window reopens, with the ELIST Database Fill Segment in the list of currently installed segments.



**Figure 17. Segment Installer Window after Installing the ELIST Database Fill Segment**

Test Status: Passed.

#### 4.2.4 ELIST Database Segment

Because it is interactive, demonstrating that the installation of the ELIST Database Segment is performed correctly requires a sequence of small tests.

##### 4.2.4.1 Test IDS-PromptForSystemPassword

Goal: Verify that the administrator is prompted to enter the previously assigned password of the SYSTEM user of the ELIST database instance.

Test Description: Follow the steps in the *ELIST IP* for installing the ELIST Database Segment, stopping when prompted to enter the SYSTEM password.

Expected Results: The Segment Installer should prompt for the password previously assigned to the SYSTEM user of the ELIST database instance (the instance called **ELIST1** here).

Test Results: The window shown in Figure 18 opens.



**Figure 18. Prompt for `SYSTEM` Password**

Test Status: Passed.

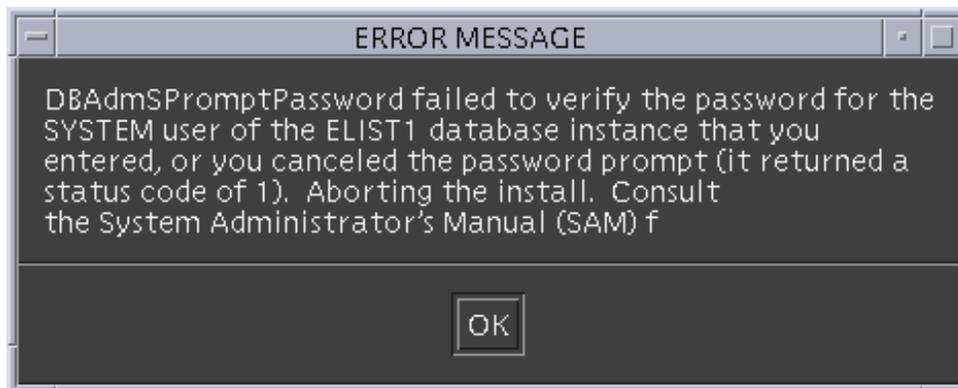
#### 4.2.4.2 Test `IDS-CancelSystemPasswordPrompt`

Goal: Verify that the installation is aborted if the administrator clicks **Cancel** instead of entering the `SYSTEM` password.

Test Description: In the currently open window, which is shown in Figure 18, leave the textbox empty and click **Cancel**.

Expected Results: The Segment Installer should abort the installation of the `ELIST` Database Segment.

Test Results: The window shown in Figure 19 opens.<sup>5</sup>



**Figure 19. Message Noting Failure to Satisfy Prompt for `SYSTEM` Password**

Upon clicking **OK**, the Segment Installer window shown in Figure 17 reopens. (The `ELIST` Database Segment is not in the list of currently installed segments.)

Test Status: Passed.

<sup>5</sup> Apparently, the text of the message is too long for the tool (`COEInstError`) used to display the message and is truncated. The last sentence of the message should read as follows: "Consult the System Administrator's Manual (SAM) for this segment."

#### 4.2.4.3 Test IDS-FailToEnterSystemPassword

Goal: Verify that the installation is aborted if, after three attempts, the administrator fails to enter the `SYSTEM` password successfully.

Test Description: Repeat Test IDS-PromptForSystemPassword in Section 4.2.4.1. In the window that opens as a result, which is shown in Figure 18, enter, three times in succession, an incorrect password, clicking **OK** after doing so each time. A message window opens to explain your error. (This behavior is programmed into the `DBAdmSPromptPassword` API of the `DBAdmS` segment, not into the `ELIST Database Segment`.) After the first two failures, click **OK** to close that window and reopen the window shown in Figure 18. After the third failure, click **OK** again.

Expected Results: The Segment Installer should abort the installation of the `ELIST Database Segment`.

Test Results: The window shown in Figure 19 opens.

Upon clicking **OK** in that window, the Segment Installer window shown in Figure 17 reopens. (The `ELIST Database Segment` is not in the list of currently installed segments.)

Test Status: Passed.

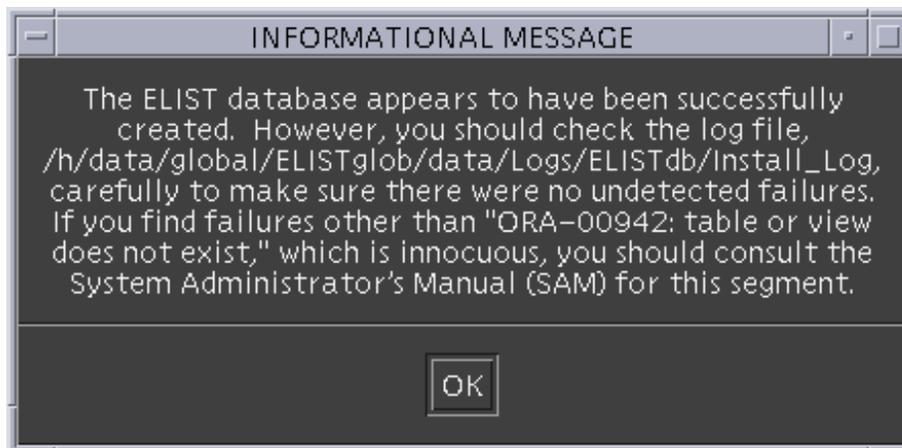
#### 4.2.4.4 Test IDS-CreateDatabase

Goal: Verify that, after successfully entering the `SYSTEM` password, the administrator is informed that the database has been created.

Test Description: Repeat Test IDS-PromptForSystemPassword in Section 4.2.4.1. In the window that opens as a result, which is shown in Figure 18, enter the correct password of the `SYSTEM` user of the `ELIST` database instance and click **OK**.

Expected Results: After a long delay, the Segment Installer should inform the administrator that the `ELIST` database has been successfully created.

Test Results: The window shown in Figure 20 opens.



**Figure 20. Message Noting Successful Creation of the `ELIST` Database**

Test Status: Passed.

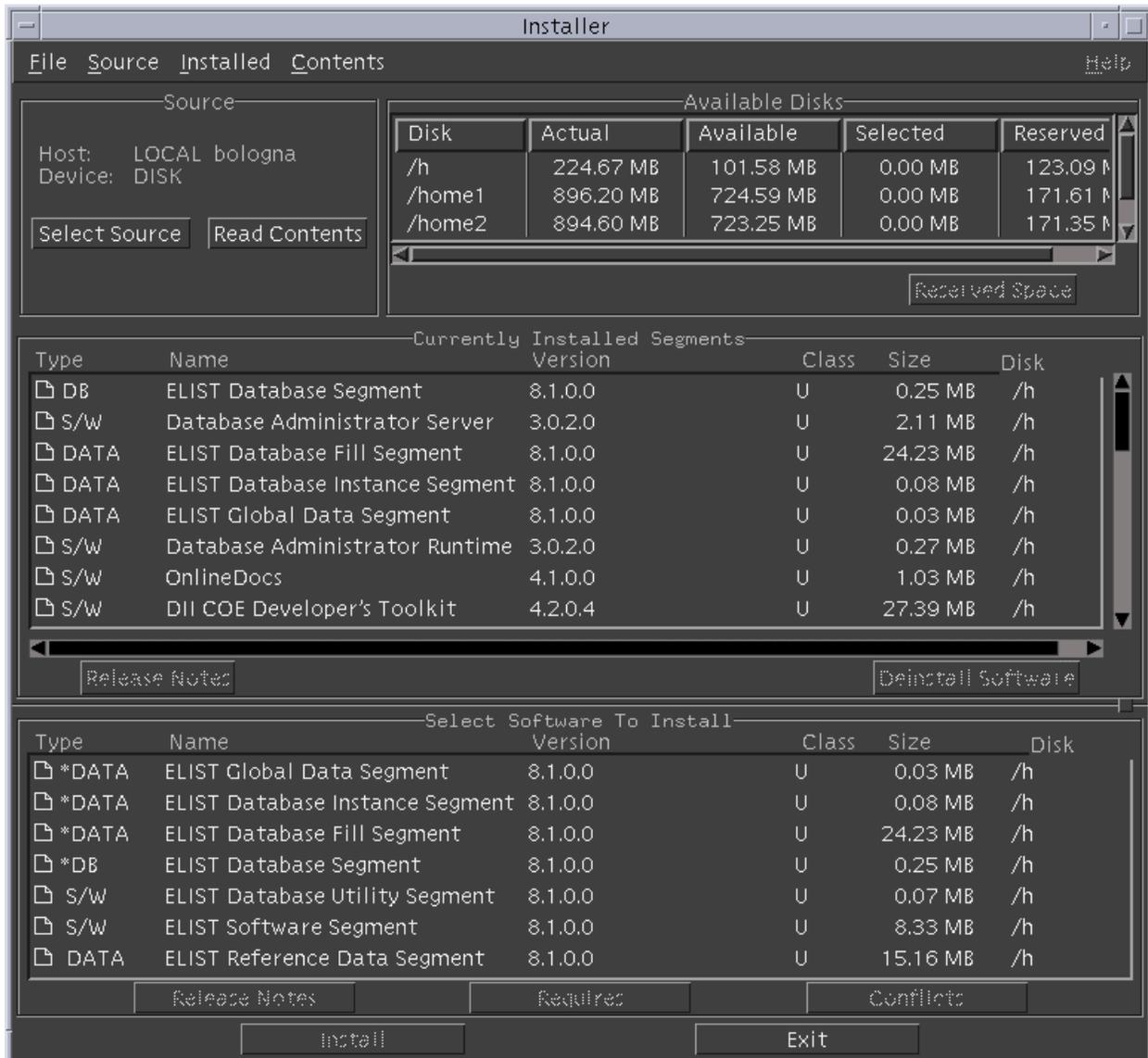
#### 4.2.4.5 Test IDS-FinishInstall

Goal: Verify that the ELIST Database Segment installs correctly.

Test Description: In the currently open window, which is shown in Figure 20, click **OK**.

Expected Results: The Segment Installer should complete the installation of the ELIST Database Segment

Test Results: As shown in Figure 21, the Segment Installer window reopens, with the ELIST Database Segment in the list of currently installed segments.



**Figure 21. Segment Installer Window after Installing the ELIST Database Segment**

Test Status: Passed.

## 4.2.5 ELIST Database Utility Segment

Because it is not interactive, demonstrating that the installation of the ELIST Database Utility Segment is performed correctly requires only one test.

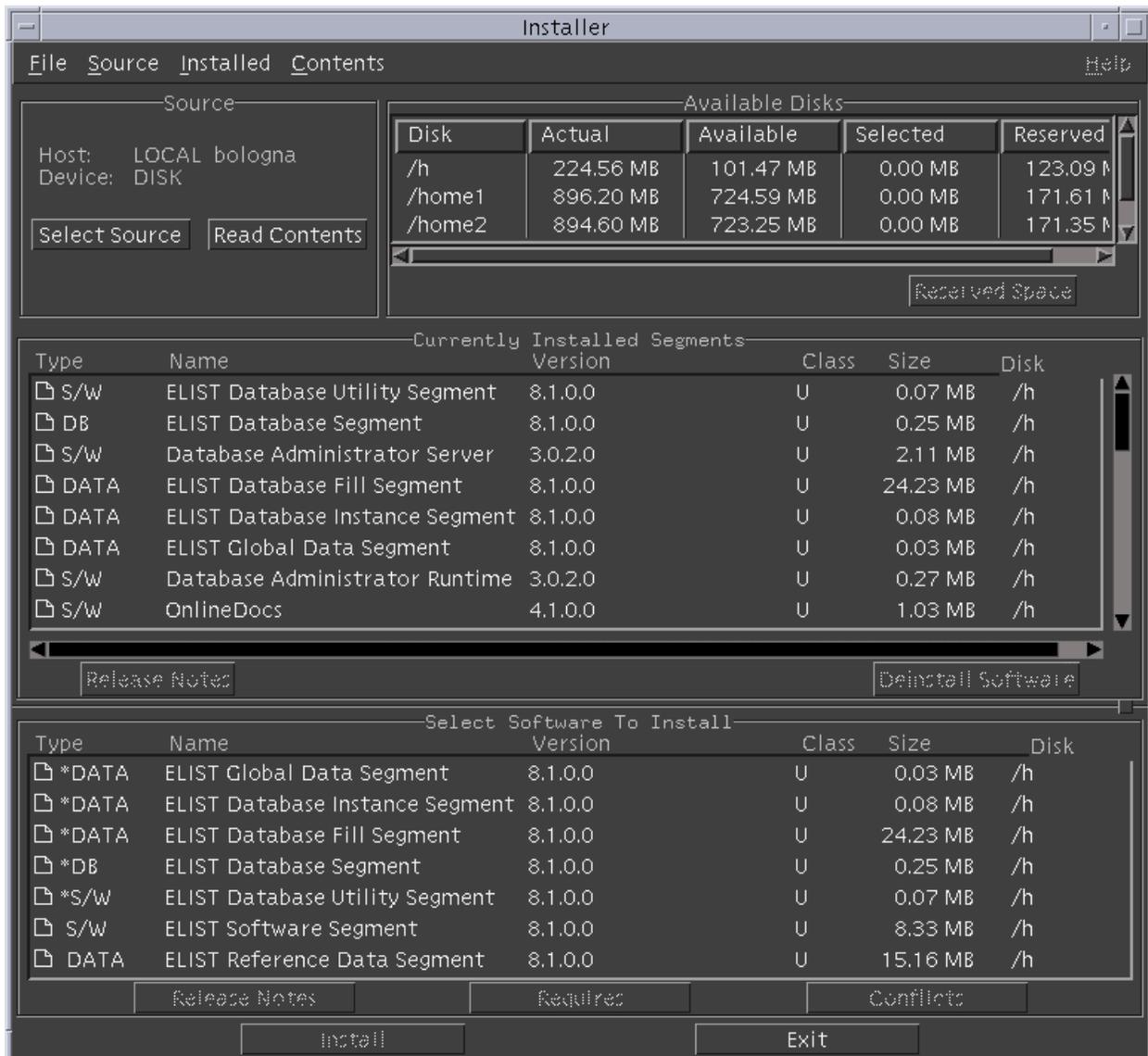
### 4.2.5.1 Test IDUS-FinishInstall

Goal: Verify that the ELIST Database Utility Segment installs correctly.

Test Description: Follow the steps in the *ELIST IP* for installing the ELIST Database Utility Segment.

Expected Results: The Segment Installer should complete the installation of the ELIST Database Utility Segment.

Test Results: As shown in Figure 22, the Segment Installer window reopens, with the ELIST Database Utility Segment in the list of currently installed segments.



**Figure 22. Segment Installer Window after Installing the ELIST Database Utility Segment**

Test Status: Passed.

#### **4.2.6 ELIST Software Segment**

Because it is not interactive, demonstrating that the installation of the ELIST Software Segment is performed correctly requires only one test on the database server, which is then repeated on an application client in preparation for later functional testing on both platforms.

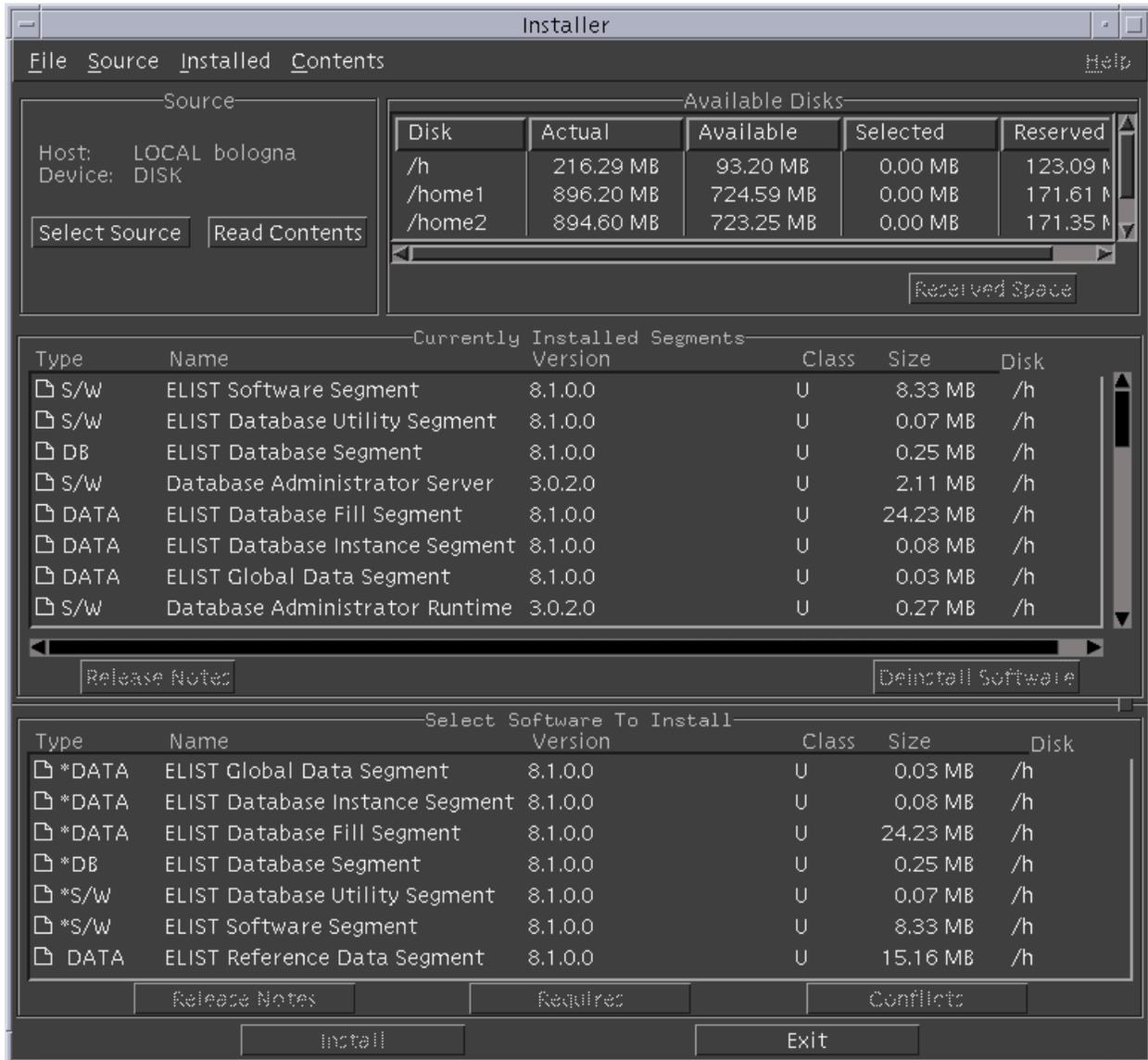
##### **4.2.6.1 Test ISS-FinishServerInstall**

Goal: Verify that the ELIST Software Segment installs correctly on the database server.

Test Description: Follow the steps in the *ELIST IP* for installing the ELIST Software Segment.

Expected Results: The Segment Installer should complete the installation of the ELIST Software Segment.

Test Results: As shown in Figure 23, the Segment Installer window reopens, with the ELIST Software Segment in the list of currently installed segments.



**Figure 23. Segment Installer Window after Installing the ELIST Software Segment on the Database Server**

Test Status: Passed.

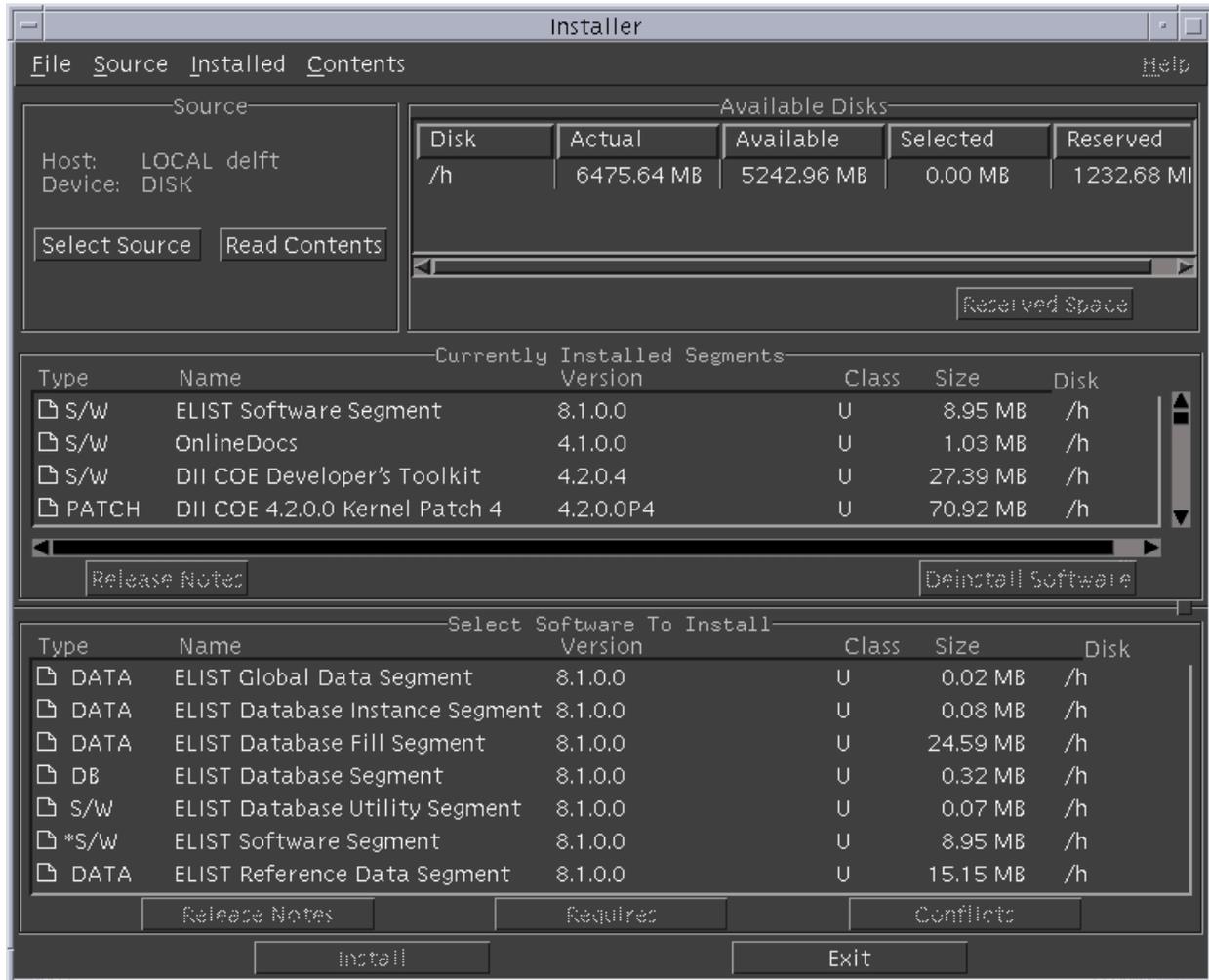
#### 4.2.6.2 Test ISS-FinishClientInstall

Goal: Verify that the ELIST Software Segment installs correctly on a separate application client.

Test Description: Log out on the database server, and log in as `sysadmin` on a separate application client. Follow the steps in the *ELIST IP* for installing the ELIST Software Segment.

Expected Results: The Segment Installer should complete the installation of the ELIST Software Segment.

Test Results: As shown in Figure 24, the Segment Installer window reopens, with the ELIST Software Segment in the list of currently installed segments.



**Figure 24. Segment Installer Window after Installing the ELIST Software Segment on the Application Client**

Test Status: Passed.

#### 4.2.7 ELIST Reference Data Segment

Because it is not interactive, demonstrating that the installation of the ELIST Reference Data Segment is performed correctly requires only one test on the database server, which is then repeated on an application client in preparation for later functional testing on both platforms.

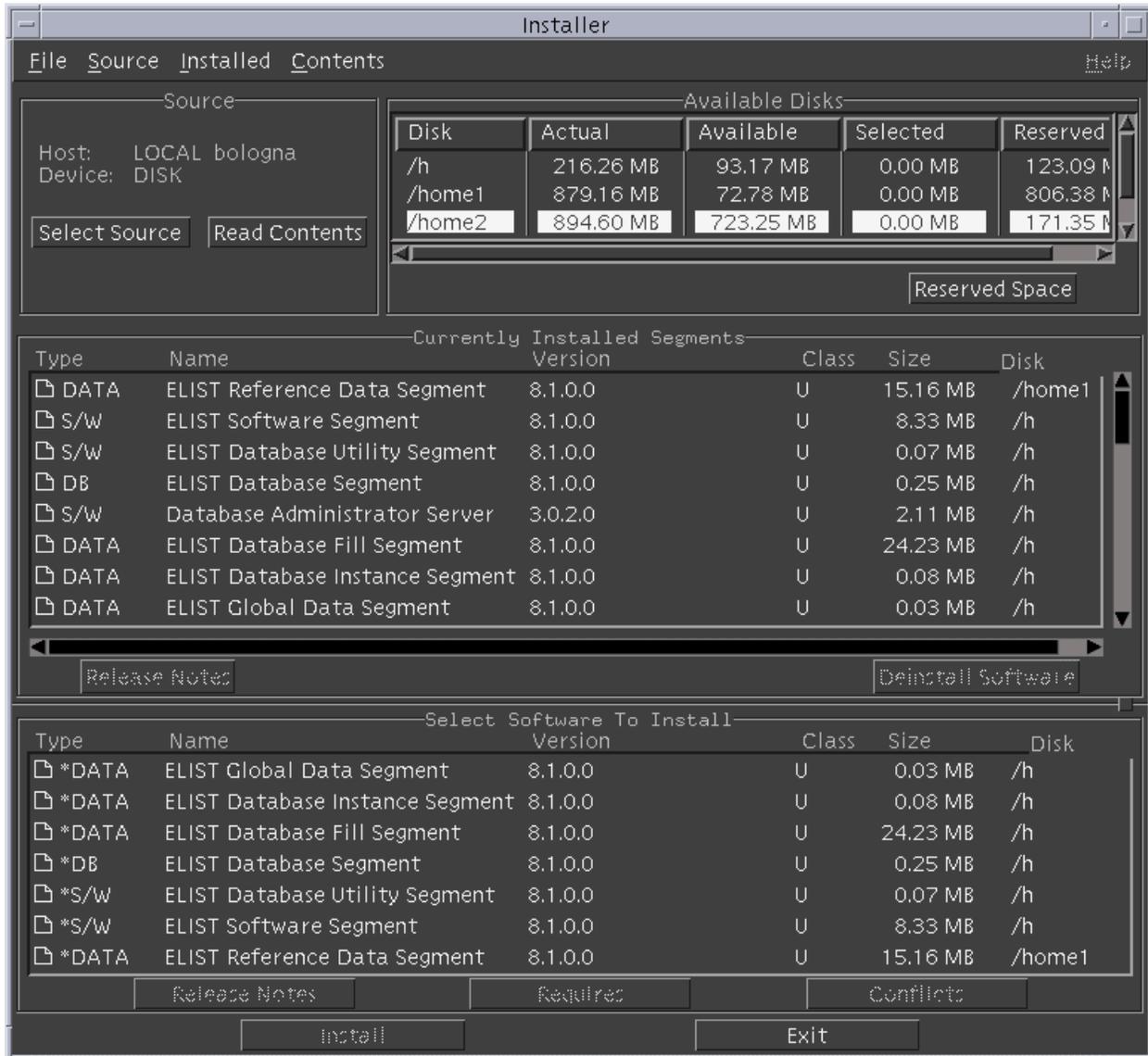
##### 4.2.7.1 Test IRDS-FinishServerInstall

Goal: Verify that the ELIST Reference Data Segment installs correctly on the database server.

Test Description: Log out on the application client, and log in again as `sysadmin` on the database server. Follow the steps in the *ELIST IP* for installing the ELIST Reference Data Segment.

Expected Results: The Segment Installer should complete the installation of the ELIST Reference Data Segment.

Test Results: As shown in Figure 25, the Segment Installer window reopens, with the ELIST Reference Data Segment in the list of currently installed segments.



**Figure 25. Segment Installer Window after Installing the ELIST Reference Data Segment on the Database Server**

Test Status: Passed

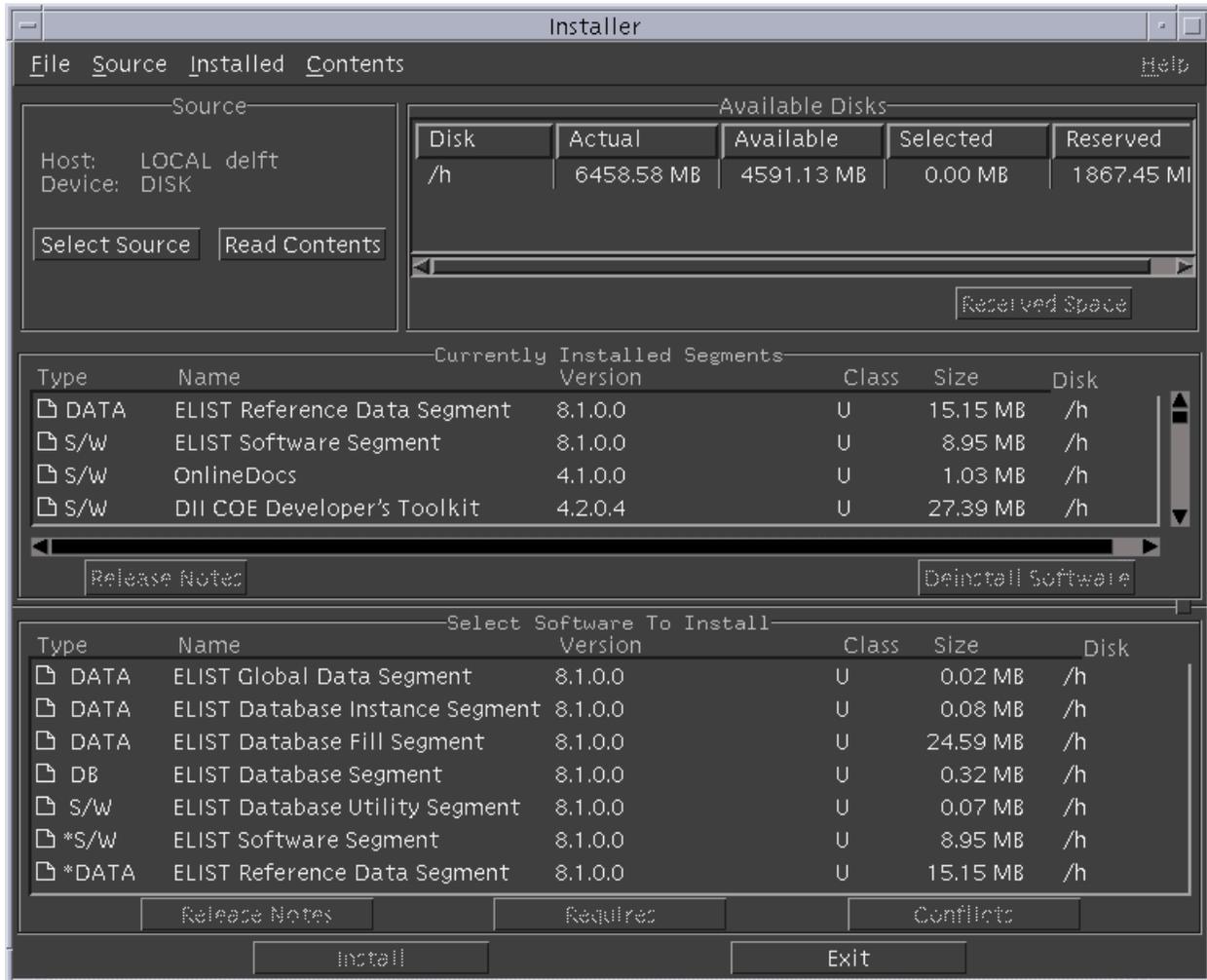
#### 4.2.7.2 Test IRDS-FinishClientInstall

Goal: Verify that the ELIST Reference Data Segment installs correctly on a separate application client.

Test Description: Log out on the database server, and log in again as `sysadmin` on a separate application client. Follow the steps in the *ELIST IP* for installing the ELIST Reference Data Segment.

Expected Results: The Segment Installer should complete the installation of the ELIST Reference Data Segment.

Test Results: As shown in Figure 26, the Segment Installer window reopens, with the ELIST Reference Data Segment in the list of currently installed segments.



**Figure 26. Segment Installer Window after Installing the ELIST Reference Data Segment on the Database Server**

Test Status: Passed.

### 4.3 Functional Tests of the Features of the ELIST Database Utility Segment

The tests in this section demonstrate that administrative ELIST users can launch the features of the ELIST Database Utility Segment.

Log out on the application client and log back in on the database server as an administrative ELIST user. (If necessary, follow the directions in the *ELIST IP* for creating the **Administrative ELIST User** profile, which accompany the installation instructions for the ELIST Database Utility Segment, and then follow the directions in the “Access Control” section of the *User’s Manual (UM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Software Segment* for creating administrative ELIST users.)

#### 4.3.1 Create User Account Feature

This feature is tested by using it to create a database user account.

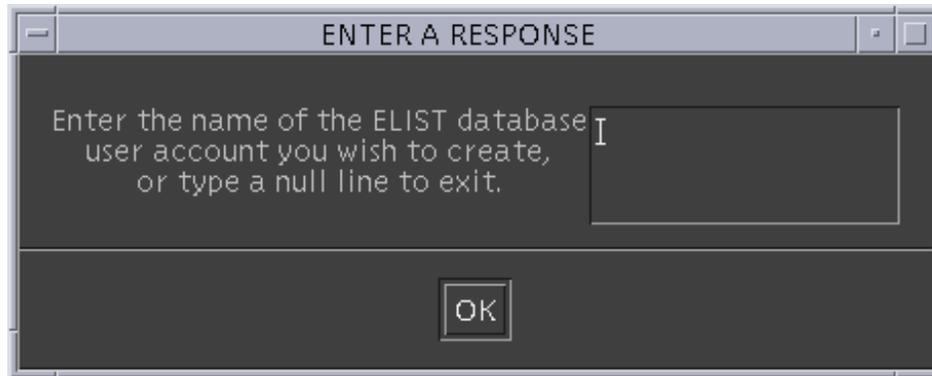
#### 4.3.1.1 Test CUA-PromptUser

Goal: Verify that the user is prompted to enter the name of the database account to create.

Test Description: Launch the **Create User Account** feature as described in the *User's Manual (UM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Utility Segment*.

Expected Results: The feature should prompt the user to enter the name of the database account to create.

Test Results: The window shown in Figure 27 opens.



**Figure 27. Window Prompting for the Database Account to Create**

Test Status: Passed.

#### 4.3.1.2 Test CUA-CancelCreate

Goal: Verify that the feature terminates if the user clicks **OK** without entering an account name.

Test Description: In the currently open window, which is shown in Figure 27, leave the textbox empty and click **OK**.

Expected Results: The feature should display a message indicating that it is exiting.

Test Results: The window shown in Figure 28 opens.



**Figure 28. Window Confirming Abort of Create User Account**

Test Status: Passed.

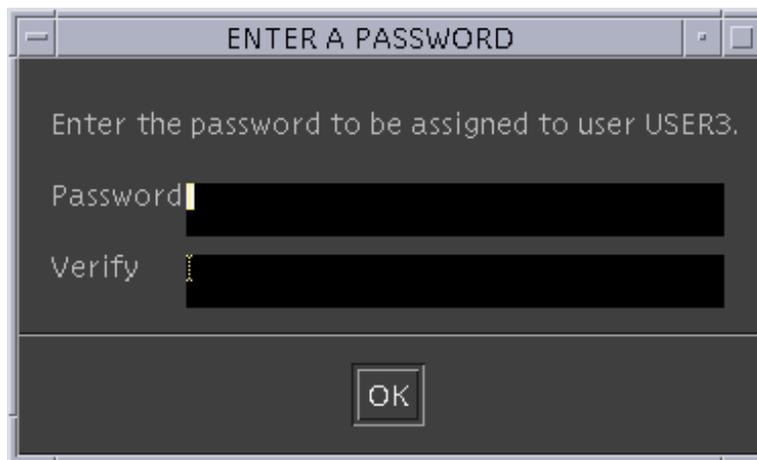
### 4.3.1.3 Test CUA-PromptUserPassword

Goal: Verify that, after entering an account name and clicking **OK**, the user is prompted to enter a password to be assigned to the new account.

Test Description: Repeat Test CUA-PromptUser in Section 4.3.1.1. In the textbox of the window that opens as a result, which is shown in Figure 27, enter **user3**, the name to be assigned to the new database account. (Normally, this feature is used to create database accounts for users who have already been given operating system accounts, but it is sufficient for the purpose of these tests to enter an arbitrary name, including one that is not an operating system userid. The only requirement, for this test, is that the name entered not be the name of an existing database account. Adjust these tests as necessary if **user3** already has a database account.) Click **OK**.

Expected Results: The feature should prompt the user to enter the password to be assigned to the new database account.

Test Results: The window shown in Figure 29 opens.



**Figure 29. Window Prompting for the Password to be Assigned to the New Database Account**

Test Status: Passed.

### 4.3.1.4 Test CUA-FailToAssignUserPassword

Goal: Verify that the feature terminates if the user fails within three attempts to enter and confirm the password to be assigned to the new account.

Test Description: In the currently open window, which is shown in Figure 29, enter, three times in succession, information *other than* the same valid password in the **Password** field and the **Verify** field, clicking **OK** after doing so each time. For example, you may:

- leave both fields, or just the **Verify** field, empty;
- enter different passwords in both fields;
- enter the same short password (less than 6 characters) in both fields; or
- enter the same long password (more than 14 characters) in both fields.

In each case, a message window opens to explain your error. (This behavior is programmed into the COEPromptPassword tool, not into the ELIST Database Utility Segment.) After the first two failures, click **OK** to close that window and reopen the window shown in Figure 29. After the third failure, click **OK** again.

Expected Results: The feature should inform the user of the failure and then terminate.

Test Results: The window shown in Figure 30 opens.



**Figure 30. Window Confirming Failure to Assign a Password to the New Database Account**

Click **OK**.

Test Status: Passed.

#### 4.3.1.5 Test CUA-PromptSystemPassword

Goal: Verify that, after successfully entering and confirming the password to be assigned to the new account, the user is prompted to enter the `SYSTEM` password.

Test Description: Repeat Test CUA-PromptUserPassword in Section 4.3.1.3. In the window that opens as a result, which is shown in Figure 29, enter the same valid password in the **Password** field and the **Verify** field (*i.e.*, a password between 6 and 14 characters in length) and click **OK**.

Expected Results: The feature should prompt the user to enter the password assigned to the `SYSTEM` user of the ELIST database instance.

Test Results: The window shown in Figure 31 opens.



**Figure 31. Window Prompting for the `SYSTEM` Password of the ELIST Database Instance**

Test Status: Passed.

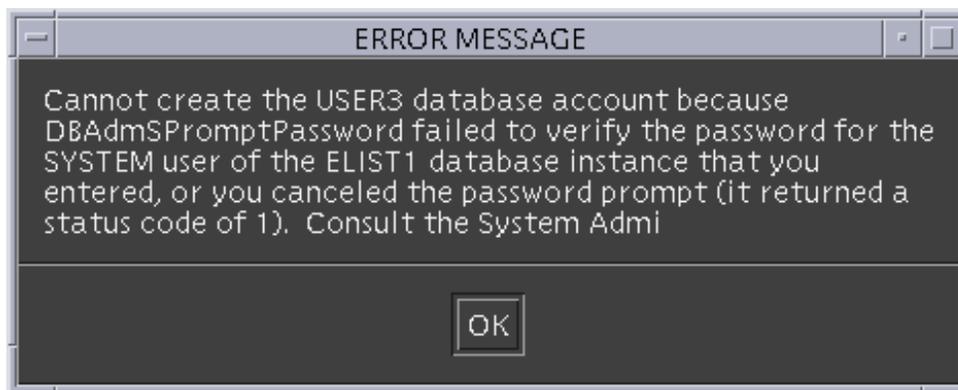
#### 4.3.1.6 Test CUA-FailToEnterSystemPassword

Goal: Verify that the feature terminates if the user fails within three attempts to enter the SYSTEM password.

Test Description: In the currently open window, which is shown in Figure 31, enter, three times in succession, an incorrect password, clicking **OK** after doing so each time. A message window opens to explain your error. (This behavior is programmed into the DBAdmSPromptPassword API of the DBAdmS segment, not into the ELIST Database Utility Segment.) After the first two failures, click **OK** to close that window and reopen the window shown in Figure 31. After the third failure, click **OK** again.

Expected Results: The feature should inform the user of the failure and terminate.

Test Results: The window shown in Figure 32 opens.<sup>6</sup>



**Figure 32. Window Confirming Failure to Enter the SYSTEM Password of the ELIST Database Instance**

Click **OK**.

Test Status: Passed.

#### 4.3.1.7 Test CUA-CreateAccount

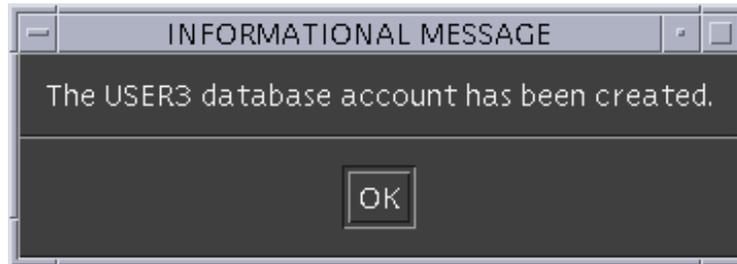
Goal: Verify that, after successfully entering the SYSTEM password, the user is informed that the database account has been created, assuming it did not initially exist.

Test Description: Repeat Test CUA-PromptSystemPassword in Section 4.3.1.5. In the window that opens as a result, which is shown in Figure 31, enter the correct password of the SYSTEM user of the ELIST database instance and click **OK**.

Expected Results: In a minute or so, the feature should display a message confirming the creation of the new database account.

Test Results: The window shown in Figure 33 opens.

<sup>6</sup> Apparently, the text of the message is too long for the tool (COEInstError) used to display the message and is truncated. The last sentence of the message should read as follows: "Consult the System Administrator's Manual (SAM) for the ELIST Database Segment" [sic].



**Figure 33. Window Confirming that the New Database Account Has Been Created**

Click **OK**.

Test Status: Passed.

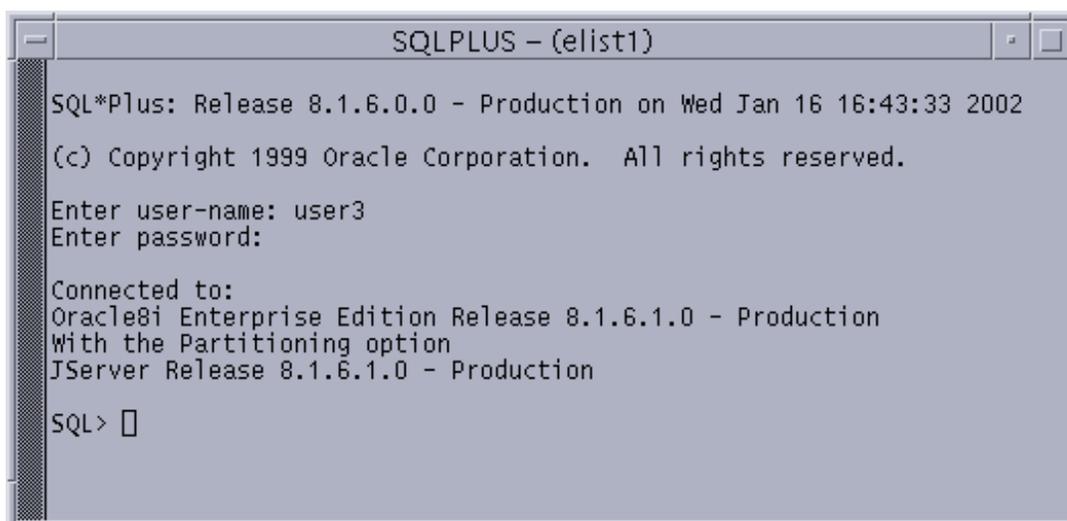
#### 4.3.1.8 Test CUA-CheckAccount

Goal: Verify that the database account now exists.

Test Description: Log out as an administrative ELIST user and log back in as a DBA (see the “Account Preparation” section of the *ELIST IP*). Launch the **SQL Plus** feature of the **ORAS** application. When a message window prompting for the “Oracle TWO\_TASK” opens, enter the name of the ELIST database instance and click **OK**. A terminal window opens next. In response to the prompt for a user-name, enter the name supplied in Test CUA-PromptUserPassword in Section 4.3.1.3 (i.e., **user3**), followed by a carriage return. In response to the prompt for that user’s password, enter the password assigned in Test CUA-PromptSystemPassword in Section 4.3.1.5, followed by a carriage return.

Expected Results: SQL\*Plus should accept the user-name and corresponding password, connect to Oracle, and prompt for an SQL command with SQL>.

Test Results: The terminal window opened by SQL\*Plus appears as shown in Figure 34 at this point.



**Figure 34. SQL\*Plus Window Showing that the Database Account Exists and Has the Correct Password**

Enter **exit**, followed by a carriage return. Log out as a DBA and log back in as an administrative ELIST user.

Test Status: Passed.

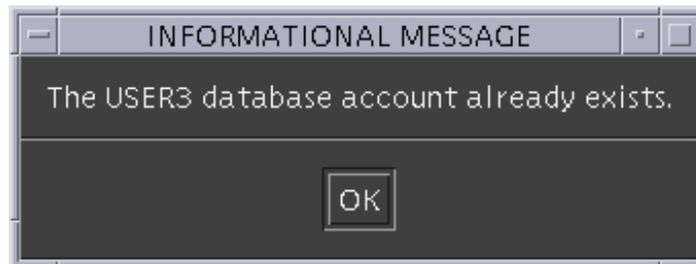
#### 4.3.1.9 Test CUA-Duplicate

Goal: Verify that, after successfully entering the `SYSTEM` password, the user is informed and the feature terminates if the database account already exists.

Test Description: Disregarding the fact that **user3** now exists, repeat Test CUA-PromptSystemPassword in Section 4.3.1.5. In the textbox of the window that opens as a result, which is shown in Figure 31, enter the correct password of the `SYSTEM` user of the ELIST database instance and click **OK**.

Expected Results: The feature should display a message indicating that account **user3** already exists.

Test Results: The window shown in Figure 35 opens.



**Figure 35. Window Reporting that the Database Account Already Exists**

Click **OK**.

Test Status: Passed.

#### 4.3.2 Remove User Account Feature

This feature is tested by using it to remove a database user account.

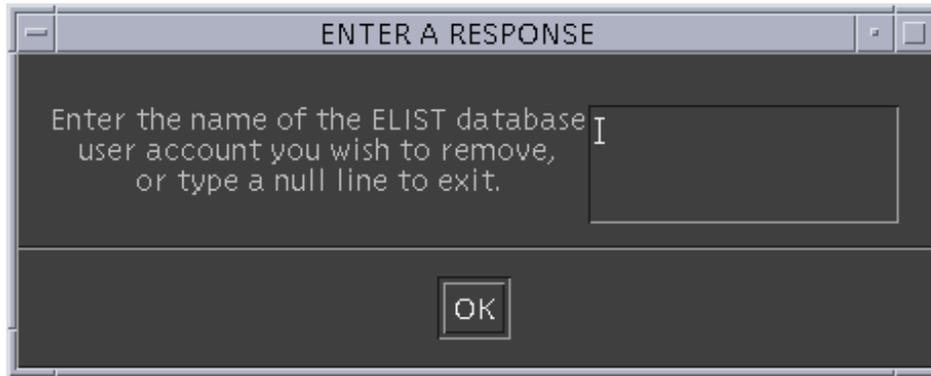
##### 4.3.2.1 Test RUA-PromptUser

Goal: Verify that the user is prompted to enter the name of the database account to remove.

Test Description: Launch the **Remove User Account** feature as described in the *User's Manual (UM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Utility Segment*.

Expected Results: The feature should prompt the user to enter the name of the database account to remove.

Test Results: The window shown in Figure 36 opens.



**Figure 36. Window Prompting for the Database Account to Remove**

Test Status: Passed.

#### **4.3.2.2 Test RUA-CancelRemove**

Goal: Verify that the feature terminates if the user clicks **OK** without entering an account name.

Test Description: In the currently open window, which is shown in Figure 36, leave the textbox empty and click **OK**.

Expected Results: The feature should display a message indicating that it is exiting.

Test Results: The window shown in Figure 37 opens.



**Figure 37. Window Confirming Abort of Remove User Account**

Test Status: Passed.

#### **4.3.2.3 Test RUA-PromptSystemPassword**

Goal: Verify that, after entering an account name and clicking **OK**, the user is prompted to enter the **SYSTEM** password.

Test Description: Repeat Test RUA-PromptUser in Section 4.3.2.1. In the textbox of the window that opens as a result, which is shown in Figure 36, enter **user3**, the name of the database account created in Test CUA-PromptUserPassword in Section 4.3.1.3.

Expected Results: The feature should prompt the user to enter the password assigned to the **SYSTEM** user of the ELIST database instance.

Test Results: The window shown in Figure 31 opens.

Test Status: Passed.

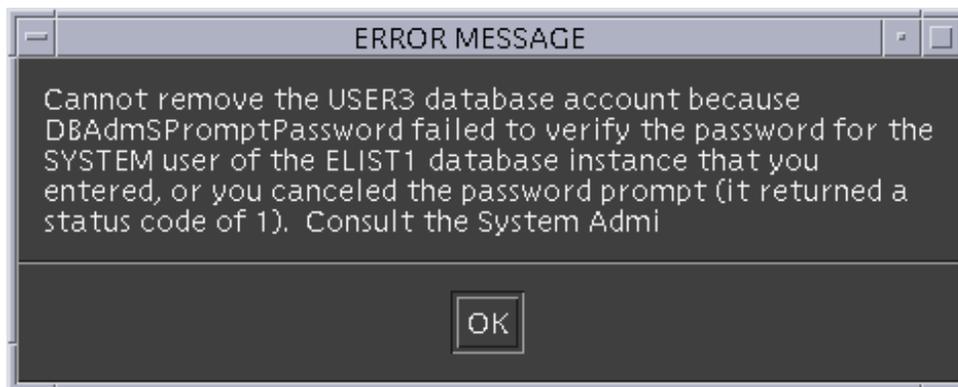
#### 4.3.2.4 Test RUA-FailToEnterSystemPassword

Goal: Verify that the feature terminates if the user fails within three attempts to enter the SYSTEM password.

Test Description: In the currently open window, which is shown in Figure 31, enter, three times in succession, an incorrect password, clicking **OK** after doing so each time. A message window opens to explain your error. (This behavior is programmed into the DBAdmSPromptPassword API of the DBAdmS segment, not into the ELIST Database Utility Segment.) After the first two failures, click **OK** to close that window and reopen the window shown in Figure 31. After the third failure, click **OK** again.

Expected Results: The feature should inform the user of the failure and terminate.

Test Results: The window shown in Figure 38 opens.<sup>7</sup>



**Figure 38. Window Confirming Failure to Enter the SYSTEM Password of the ELIST Database Instance**

Click **OK**.

Test Status: Passed.

#### 4.3.2.5 Test RUA-RemoveAccount

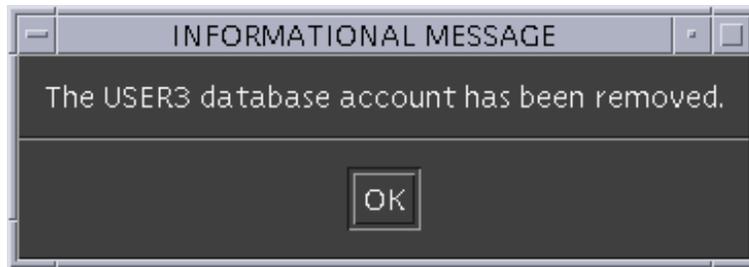
Goal: Verify that, after successfully entering the SYSTEM password, the user is informed that the database account has been removed, assuming the account initially existed.

Test Description: Repeat Test RUA-PromptSystemPassword in Section 4.3.2.3. In the window that opens as a result, which is shown in Figure 31, enter the correct password of the SYSTEM user of the ELIST database instance and click **OK**.

Expected Results: In a minute or so, the feature should display a message confirming the removal of the database account.

<sup>7</sup> Apparently, the text of the message is too long for the tool (COEInstError) used to display the message and is truncated. The last sentence of the message should read as follows: "Consult the System Administrator's Manual (SAM) for the ELIST Database Segment" [*sic*].

Test Results: The window shown in Figure 39 opens.



**Figure 39. Window Confirming that the Database Account Has Been Removed**

Click **OK**.

Test Status: Passed.

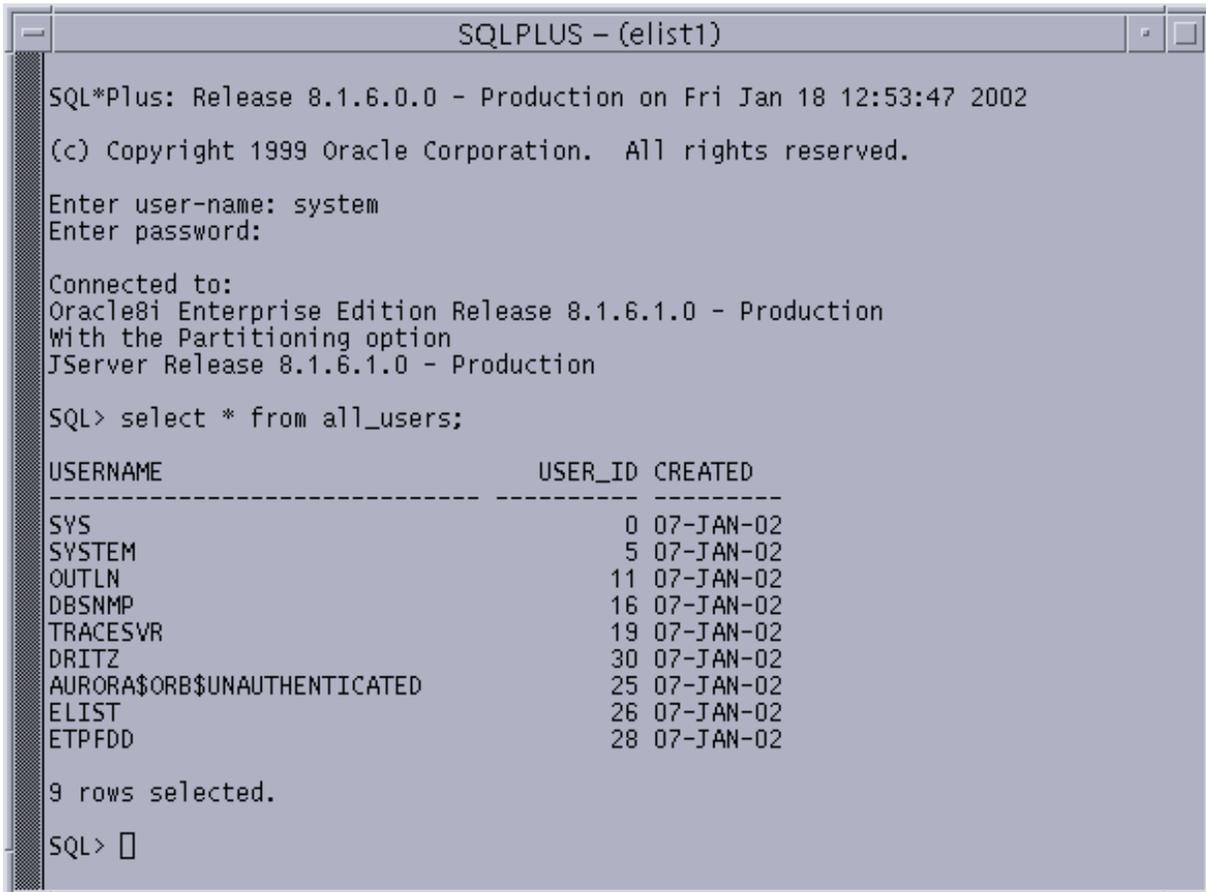
#### **4.3.2.6 Test RUA-CheckAccount**

Goal: Verify that the database account no longer exists.

Test Description: Log out as an administrative ELIST user and log back in as a DBA (see the “Account Preparation” section of the *ELIST IP*). Launch the **SQL Plus** feature of the **ORAS** application. When a message window prompting for the “Oracle TWO\_TASK” opens, enter the name of the ELIST database instance and click **OK**. A terminal window opens next. In response to the prompt for a user-name, enter **system**. In response to the prompt for the password, enter the password of the **SYSTEM** user of the ELIST database instance. The **SYSTEM** user is connected to Oracle, and SQL\*Plus prompts for an SQL command. In response to that prompt, enter **select \* from all\_users;** followed by a carriage return.

Expected Results: The list of usernames displayed by SQL\*Plus should not include **user3**.

Test Results: The terminal window opened by SQL\*Plus appears as shown in Figure 40 at this point.



```

SQL*Plus: Release 8.1.6.0.0 - Production on Fri Jan 18 12:53:47 2002
(c) Copyright 1999 Oracle Corporation. All rights reserved.

Enter user-name: system
Enter password:

Connected to:
Oracle8i Enterprise Edition Release 8.1.6.1.0 - Production
With the Partitioning option
JServer Release 8.1.6.1.0 - Production

SQL> select * from all_users;

-----
USERNAME                                USER_ID  CREATED
-----
SYS                                       0 07-JAN-02
SYSTEM                                   5 07-JAN-02
OUTLN                                    11 07-JAN-02
DBSNMP                                  16 07-JAN-02
TRACESVR                                19 07-JAN-02
DRITZ                                    30 07-JAN-02
AURORA$ORB$UNAUTHENTICATED             25 07-JAN-02
ELIST                                    26 07-JAN-02
ETPFDD                                   28 07-JAN-02

9 rows selected.

SQL> 

```

**Figure 40. SQL\*Plus Window Showing that the Database Account No Longer Exists**

Enter **exit** and type a carriage return. Log out as a DBA and log back in as an administrative ELIST user.

Test Status: Passed.

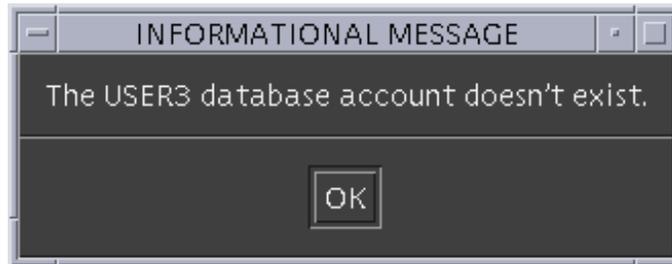
#### 4.3.2.7 Test RUA-Nonexistent

Goal: Verify that, after successfully entering the **SYSTEM** password, the user is informed and the feature terminates if the database account does not exist.

Test Description: Disregarding the fact that **user3** no longer exists, repeat Test RUA-PromptSystemPassword in Section 4.3.2.3. In the window that opens as a result, which is shown in Figure 31, enter the correct password of the **SYSTEM** user of the ELIST database instance and click **OK**.

Expected Results: The feature should display a message indicating that account **user3** does not exist.

Test Results: The window shown in Figure 41 opens.



**Figure 41. Window Reporting that the Database Account Does Not Exist**

Click **OK**. Recreate the database account for **user3** by repeating Test CUA-CreateAccount in Section 4.3.1.7.

Test Status: Passed.

### 4.3.3 Lock User Account Feature

This feature is tested by using it to lock a database user account.

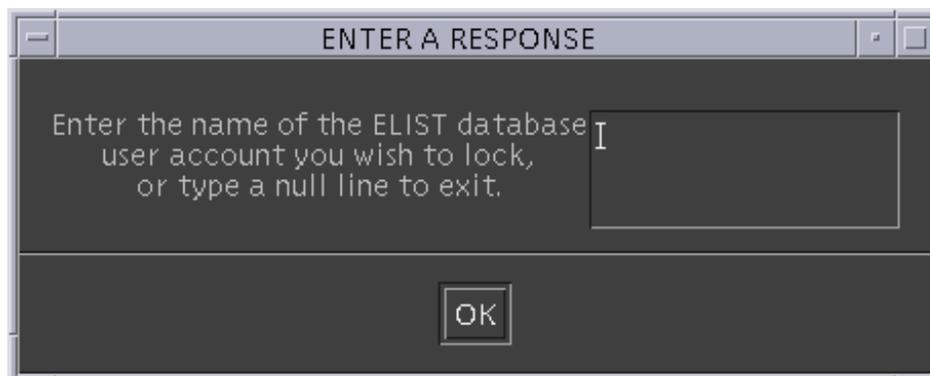
#### 4.3.3.1 Test LoUA-PromptUser

Goal: Verify that the user is prompted to enter the name of the database account to lock.

Test Description: Launch the **Lock User Account** feature as described in the *User's Manual (UM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Utility Segment*.

Expected Results: The feature should prompt the user to enter the name of the database account to lock.

Test Results: The window shown in Figure 42 opens.



**Figure 42. Window Prompting for the Database Account to Lock**

Test Status: Passed.

#### 4.3.3.2 Test LoUA-CancelLock

Goal: Verify that the feature terminates if the user clicks **OK** without entering an account name.

Test Description: In the currently open window, which is shown in Figure 42, leave the textbox empty and click **OK**.

Expected Results: The feature should display a message indicating that it is exiting.

Test Results: The window shown in Figure 43 opens.



**Figure 43. Window Confirming Abort of Lock User Account**

Test Status: Passed.

#### 4.3.3.3 Test LoUA-PromptSystemPassword

Goal: Verify that, after entering an account name and clicking **OK**, the user is prompted to enter the SYSTEM password.

Test Description: Repeat Test LoUA-PromptUser in Section 4.3.3.1. In the textbox of the window that opens as a result, which is shown in Figure 42, enter **user3**, the name of the database account created at the end of Test RUA-Nonexistent in Section 4.3.2.7.

Expected Results: The feature should prompt the user to enter the password assigned to the SYSTEM user of the ELIST database instance.

Test Results: The window shown in Figure 31 opens.

Test Status: Passed.

#### 4.3.3.4 Test LoUA-FailToEnterSystemPassword

Goal: Verify that the feature terminates if the user fails within three attempts to enter the SYSTEM password.

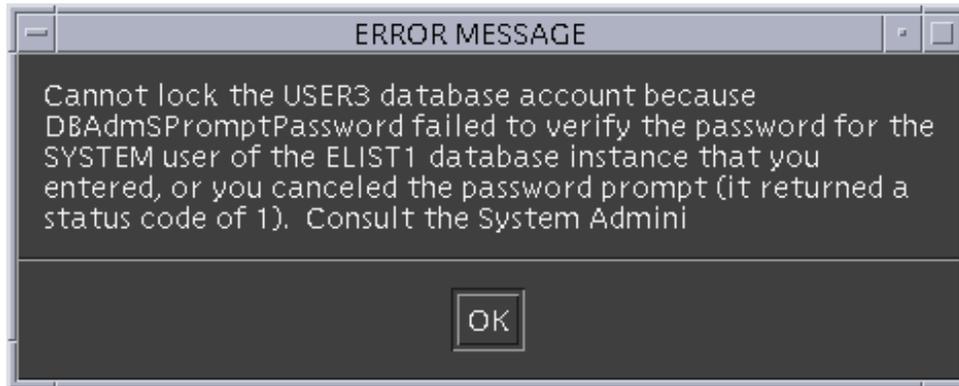
Test Description: In the currently open window, which is shown in Figure 31, enter, three times in succession, an incorrect password, clicking **OK** after doing so each time. A message window opens to explain your error. (This behavior is programmed into the DBAdmSPromptPassword API of the DBAdmS segment, not into the ELIST Database Utility Segment.) After the first two failures, click **OK** to close that window and reopen the window shown in Figure 31. After the third failure, click **OK** again.

Expected Results: The feature should inform the user of the failure and terminate.

Test Results: The window shown in Figure 44 opens.<sup>8</sup>

---

<sup>8</sup> Apparently, the text of the message is too long for the tool (COEInstError) used to display the message and is truncated. The last sentence of the message should read as follows: "Consult the System Administrator's Manual (SAM) for the ELIST Database Segment" [*sic*].



**Figure 44. Window Confirming Failure to Enter the **SYSTEM** Password of the **ELIST** Database Instance**

Click **OK**.

Test Status: Passed.

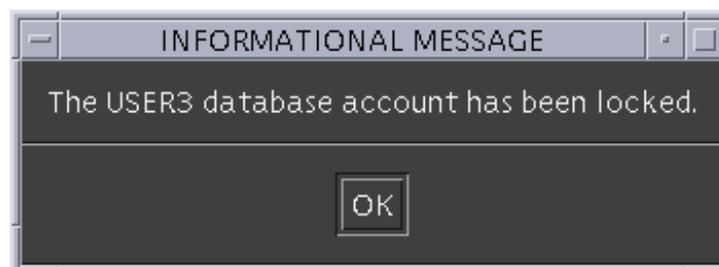
#### **4.3.3.5 Test LoUA-LockAccount**

Goal: Verify that, after successfully entering the **SYSTEM** password, the user is informed that the database account has been locked, assuming the account exists.

Test Description: Repeat Test LoUA-PromptSystemPassword in Section 4.3.3.3. In the window that opens as a result, which is shown in Figure 31, enter the correct password of the **SYSTEM** user of the **ELIST** database instance and click **OK**.

Expected Results: In a minute or so, the feature should display a message confirming the locking of the database account.

Test Results: The window shown in Figure 45 opens.



**Figure 45. Window Confirming that the Database Account Has Been Locked**

Click **OK**.

Test Status: Passed.

#### **4.3.3.6 Test LoUA-CheckAccount**

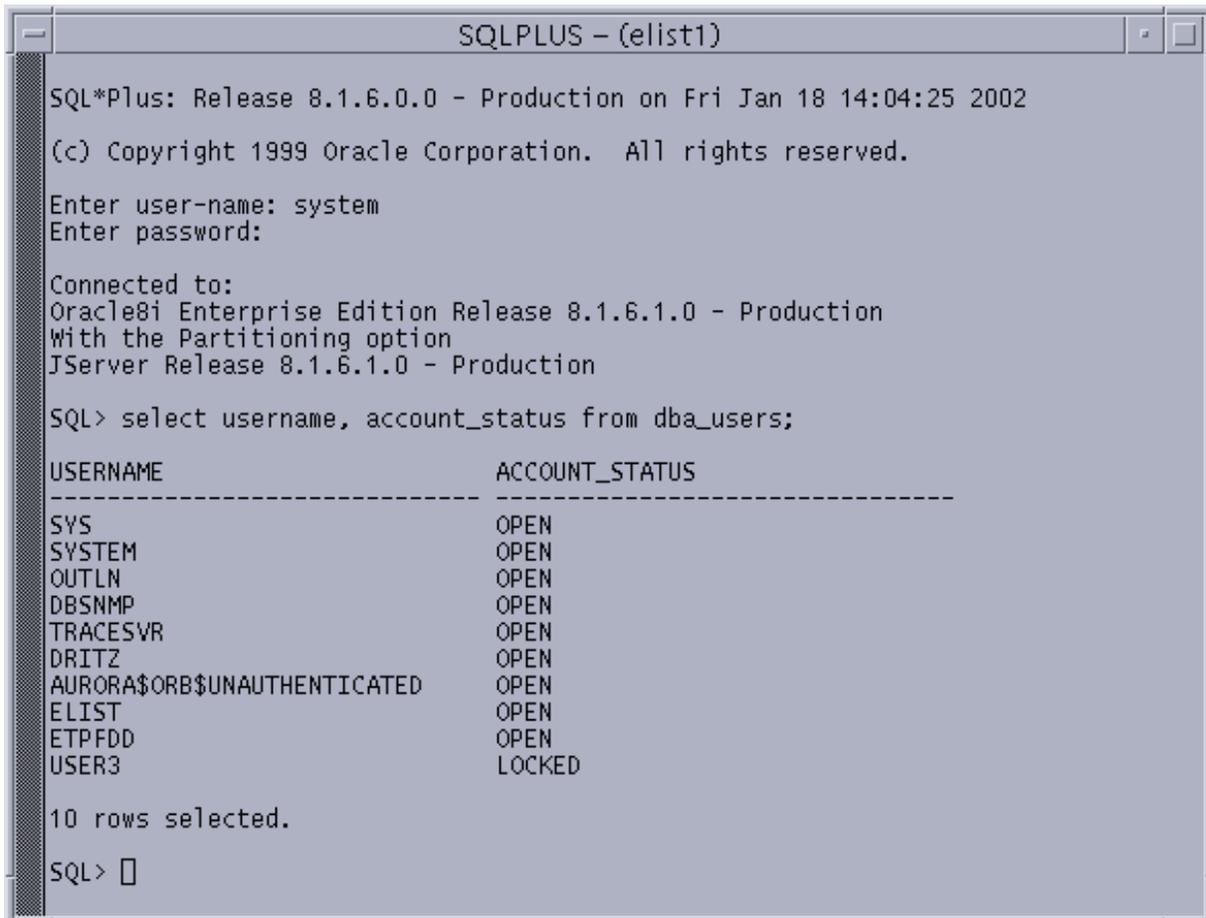
Goal: Verify that the database account has been locked.

Test Description: Log out as an administrative **ELIST** user and log back in as a **DBA** (see the “Account Preparation” section of the *ELIST IP*). Launch the **SQL Plus** feature of the **ORAS**

application. When a message window prompting for the “Oracle TWO\_TASK” opens, enter the name of the ELIST database instance and click **OK**. A terminal window opens next. In response to the prompt for a user-name, enter **system**. In response to the prompt for the password, enter the password of the **SYSTEM** user of the ELIST database instance. The **SYSTEM** user is connected to Oracle, and SQL\*Plus prompts for an SQL command. In response to that prompt, enter **select username, account\_status from dba\_users;** followed by a carriage return.

Expected Results: The list of usernames and statuses displayed by SQL\*Plus should show that account **user3** exists but is locked.

Test Results: The terminal window opened by SQL\*Plus appears as shown in Figure 46 at this point.



```

SQLPLUS - (elist1)
SQL*Plus: Release 8.1.6.0.0 - Production on Fri Jan 18 14:04:25 2002
(c) Copyright 1999 Oracle Corporation. All rights reserved.
Enter user-name: system
Enter password:
Connected to:
Oracle8i Enterprise Edition Release 8.1.6.1.0 - Production
With the Partitioning option
JServer Release 8.1.6.1.0 - Production
SQL> select username, account_status from dba_users;
-----
USERNAME                                ACCOUNT_STATUS
-----
SYS                                       OPEN
SYSTEM                                  OPEN
OUTLN                                    OPEN
DBSNMP                                   OPEN
TRACESVR                                OPEN
DRITZ                                    OPEN
AURORA$ORB$UNAUTHENTICATED              OPEN
ELIST                                    OPEN
ETPFDD                                   OPEN
USER3                                    LOCKED

10 rows selected.

SQL>

```

**Figure 46. SQL\*Plus Window Showing that the Database Account Has Been Locked**

Enter **exit** and type a carriage return. Log out as a DBA and log back in as an administrative ELIST user.

Test Status: Passed.

#### 4.3.3.7 Test LoUA-Nonexistent

Goal: Verify that, after successfully entering the **SYSTEM** password, the user is informed and the feature terminates if the database account does not exist.

Test Description: Remove the database account for **user3** by repeating Test RUA-RemoveAccount in Section 4.3.2.5. Disregarding the fact that **user3** no longer exists, repeat Test LoUA-PromptSystemPassword in Section 4.3.3.3. In the window that opens as a result, which is shown in Figure 31, enter the correct password of the SYSTEM user of the ELIST database instance and click **OK**.

Expected Results: The feature should display a message indicating that account **user3** does not exist.

Test Results: The window shown in Figure 41 opens.

Click **OK**. Recreate the database account for **user3** by repeating Test CUA-CreateAccount in Section 4.3.1.7.

Test Status: Passed.

#### 4.3.4 Unlock User Account Feature

This feature is tested by using it to unlock a database user account.

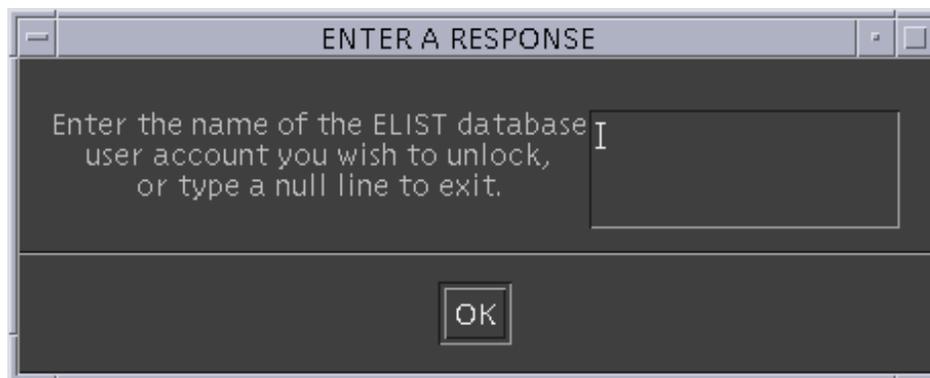
##### 4.3.4.1 Test UUA-PromptUser

Goal: Verify that the user is prompted to enter the name of the database account to unlock.

Test Description: Launch the **Unlock User Account** feature as described in the *User's Manual (UM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Utility Segment*.

Expected Results: The feature should prompt the user to enter the name of the database account to unlock.

Test Results: The window shown in Figure 47 opens.



**Figure 47. Window Prompting for the Database Account to Unlock**

Test Status: Passed.

##### 4.3.4.2 Test UUA-CancelUnlock

Goal: Verify that the feature terminates if the user clicks **OK** without entering an account name.

Test Description: In the currently open window, which is shown in Figure 47, leave the textbox empty and click **OK**.

Expected Results: The feature should display a message indicating that it is exiting.

Test Results: The window shown in Figure 48 opens.



**Figure 48. Window Confirming Abort of Unlock User Account**

Test Status: Passed.

#### **4.3.4.3 Test UUA-PromptSystemPassword**

Goal: Verify that, after entering an account name and clicking **OK**, the user is prompted to enter the SYSTEM password.

Test Description: Repeat Test UUA-PromptUser in Section 4.3.4.1. In the textbox of the window that opens as a result, which is shown in Figure 47, enter **user3**, the name of the database account created at the end of Test LoUA-Nonexistent in Section 4.3.3.7.

Expected Results: The feature should prompt the user to enter the password assigned to the SYSTEM user of the ELIST database instance.

Test Results: The window shown in Figure 31 opens.

Test Status: Passed.

#### **4.3.4.4 Test UUA-FailToEnterSystemPassword**

Goal: Verify that the feature terminates if the user fails within three attempts to enter the SYSTEM password.

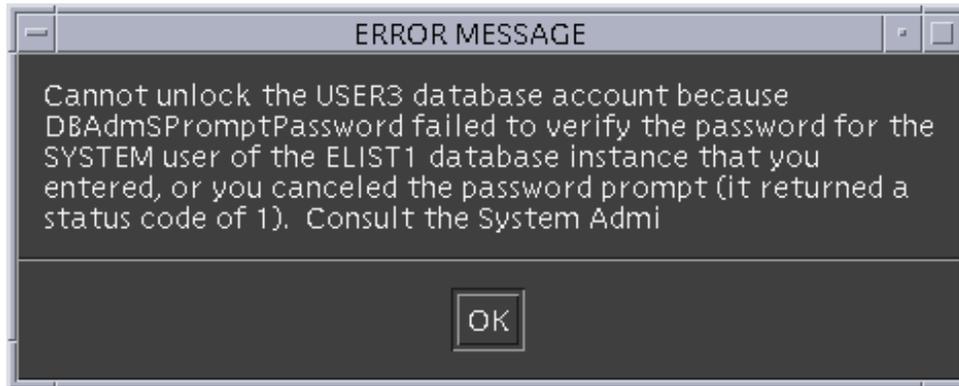
Test Description: In the currently open window, which is shown in Figure 31, enter, three times in succession, an incorrect password, clicking **OK** after doing so each time. A message window opens to explain your error. (This behavior is programmed into the DBAdmSPromptPassword API of the DBAdmS segment, not into the ELIST Database Utility Segment.) After the first two failures, click **OK** to close that window and reopen the window shown in Figure 31. After the third failure, click **OK** again.

Expected Results: The feature should inform the user of the failure and terminate.

Test Results: The window shown in Figure 49 opens.<sup>9</sup>

---

<sup>9</sup> Apparently, the text of the message is too long for the tool (COEInstError) used to display the message and is truncated. The last sentence of the message should read as follows: "Consult the System Administrator's Manual (SAM) for the ELIST Database Segment" [*sic*].



**Figure 49. Window Confirming Failure to Enter the **SYSTEM** Password of the **ELIST** Database Instance**

Click **OK**.

Test Status: Passed.

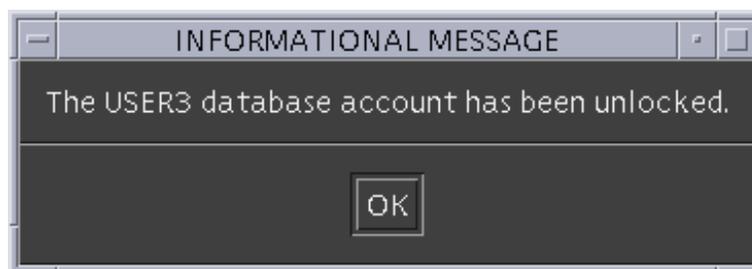
#### 4.3.4.5 Test UUA-UnlockAccount

Goal: Verify that, after successfully entering the **SYSTEM** password, the user is informed that the database account has been unlocked, assuming the account exists.

Test Description: Repeat Test UUA-PromptSystemPassword in Section 4.3.4.3. In the window that opens as a result, which is shown in Figure 31, enter the correct password of the **SYSTEM** user of the **ELIST** database instance and click **OK**.

Expected Results: In a minute or so, the feature should display a message confirming the unlocking of the database account.

Test Results: The window shown in Figure 50 opens.



**Figure 50. Window Confirming that the Database Account Has Been Unlocked**

Click **OK**.

Test Status: Passed.

#### 4.3.4.6 Test UUA-CheckAccount

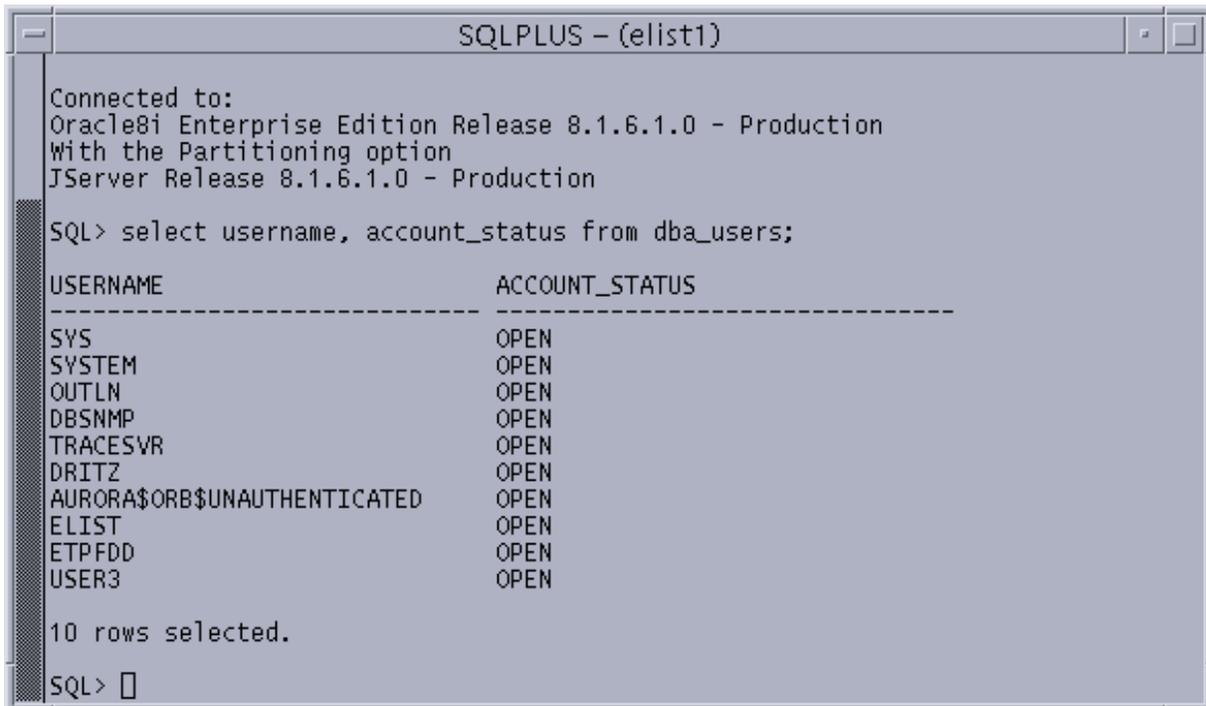
Goal: Verify that the database account has been unlocked.

Test Description: Log out as an administrative **ELIST** user and log back in as a **DBA** (see the “Account Preparation” section of the *ELIST IP*). Launch the **SQL Plus** feature of the **ORAS**

application. When a message window prompting for the “Oracle TWO\_TASK” opens, enter the name of the ELIST database instance and click **OK**. A terminal window opens next. In response to the prompt for a user-name, enter **system**. In response to the prompt for the password, enter the password of the SYSTEM user of the ELIST database instance. The SYSTEM user is connected to Oracle, and SQL\*Plus prompts for an SQL command. In response to that prompt, enter **select username, account\_status from dba\_users**; followed by a carriage return.

Expected Results: The list of usernames and statuses displayed by SQL\*Plus should show that account **user3** exists and is unlocked.

Test Results: The terminal window opened by SQL\*Plus appears as shown in Figure 51 at this point.



```

SQLPLUS - (elist1)
Connected to:
Oracle8i Enterprise Edition Release 8.1.6.1.0 - Production
With the Partitioning option
JServer Release 8.1.6.1.0 - Production

SQL> select username, account_status from dba_users;

-----
USERNAME                                ACCOUNT_STATUS
-----
SYS                                       OPEN
SYSTEM                                   OPEN
OUTLN                                    OPEN
DBSNMP                                   OPEN
TRACESVR                                 OPEN
DRITZ                                    OPEN
AURORA$ORB$UNAUTHENTICATED              OPEN
ELIST                                    OPEN
ETPFDD                                   OPEN
USER3                                    OPEN

10 rows selected.

SQL>

```

**Figure 51. SQL\*Plus Window Showing that the Database Account Has Been Unlocked**

Enter **exit** and type a carriage return. Log out as a DBA and log back in as an administrative ELIST user.

Test Status: Passed.

#### 4.3.4.7 Test UUA-Nonexistent

Goal: Verify that, after successfully entering the SYSTEM password, the user is informed and the feature terminates if the database account does not exist.

Test Description: Remove the database account for **user3** by repeating Test RUA-RemoveAccount in Section 4.3.2.5. Disregarding the fact that **user3** no longer exists, repeat Test UUA-PromptSystemPassword in Section 4.3.4.3. In the window that opens as a result, which is shown in Figure 31, enter the correct password of the SYSTEM user of the ELIST database instance and click **OK**.

Expected Results: The feature should display a message indicating that account **user3** does not exist.

Test Results: The window shown in Figure 41 opens.

Click **OK**.

Test Status: Passed.

### **4.3.5 List User Accounts Feature**

This feature is tested by using it to list database user accounts.

#### **4.3.5.1 Test LiUA-PromptSystemPassword**

Goal: Verify that the user is prompted to enter the `SYSTEM` password.

Test Description: Launch the **List User Accounts** feature as described in the *User's Manual (UM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Utility Segment*.

Expected Results: The feature should prompt the user to enter the password assigned to the `SYSTEM` user of the ELIST database instance.

Test Results: The window shown in Figure 31 opens.

Test Status: Passed.

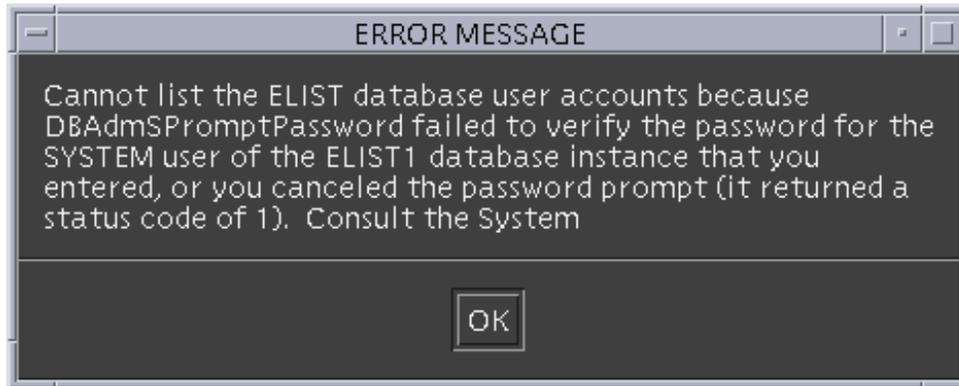
#### **4.3.5.2 Test LiUA-FailToEnterSystemPassword**

Goal: Verify that the feature terminates if the user fails within three attempts to enter the `SYSTEM` password.

Test Description: In the currently open window, which is shown in Figure 31, enter, three times in succession, an incorrect password, clicking **OK** after doing so each time. A message window opens to explain your error. (This behavior is programmed into the `DBAdmSPromptPassword` API of the `DBAdmS` segment, not into the ELIST Database Utility Segment.) After the first two failures, click **OK** to close that window and reopen the window shown in Figure 31. After the third failure, click **OK** again.

Expected Results: The feature should inform the user of the failure and terminate.

Test Results: The window shown in Figure 52 opens.



**Figure 52. Window Confirming Failure to Enter the **SYSTEM** Password of the ELIST Database Instance**

Click **OK**.

Test Status: Passed.

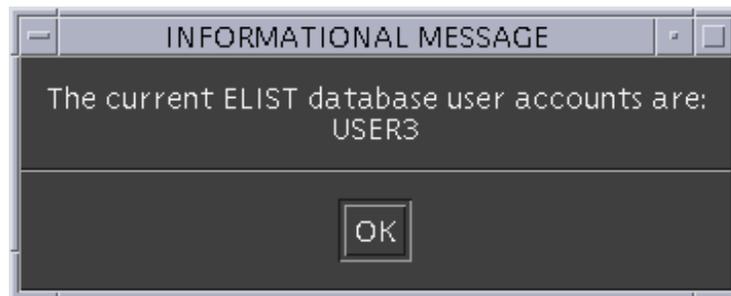
#### **4.3.5.3 Test LiUA-ListAccounts**

Goal: Verify that, after successfully entering the **SYSTEM** password, the user is presented with a list of existing accounts.

Test Description: Recreate the database account for **user3** by repeating Test CUA-CreateAccount in Section 4.3.1.7. Repeat Test LiUA-PromptSystemPassword in Section 4.3.5.1. In the window that opens as a result, which is shown in Figure 31, enter the correct password of the **SYSTEM** user of the ELIST database instance and click **OK**.

Expected Results: The feature should display a window containing a list of existing database user accounts, which should contain only **user3**.

Test Results: The window shown in Figure 53 opens.



**Figure 53. Window Showing the Existing Database Accounts**

Had no database accounts existed, the window shown in Figure 53 would have contained an empty list.

Test Status: Passed.

#### **4.3.6 Change User Password Feature**

This feature is tested by using it to change the password of a database user account.

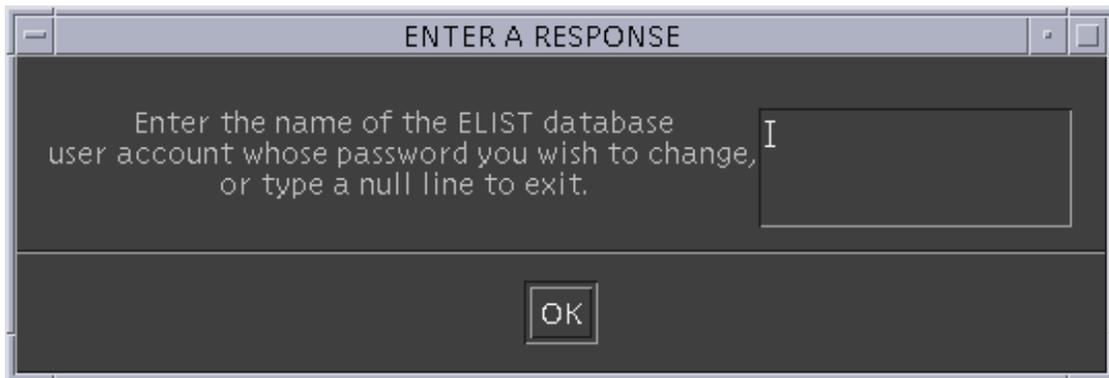
#### 4.3.6.1 Test CUP-PromptUser

Goal: Verify that the user is prompted to enter the name of the database account whose password is to be changed.

Test Description: Launch the **Change User Password** feature as described in the *User's Manual (UM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Utility Segment*.

Expected Results: The feature should prompt the user to enter the name of the database account whose password is to be changed.

Test Results: The window shown in Figure 54 opens.



**Figure 54. Window Prompting for the Database Account Whose Password is to be Changed**

Test Status: Passed.

#### 4.3.6.2 Test CUP-CancelChange

Goal: Verify that the feature terminates if the user clicks **OK** without entering an account name.

Test Description: In the currently open window, which is shown in Figure 54, leave the textbox empty and click **OK**.

Expected Results: The feature should display a message indicating that it is exiting.

Test Results: The window shown in Figure 55 opens.



**Figure 55. Window Confirming Abort of Change User Password**

Test Status: Passed.

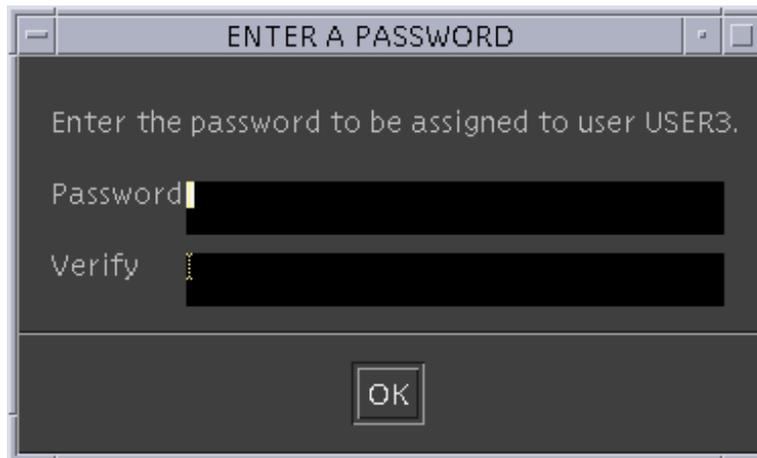
#### 4.3.6.3 Test CUP-PromptUserPassword

Goal: Verify that, after entering an account name and clicking **OK**, the user is prompted to enter a new password to be assigned to the account.

Test Description: Repeat Test CUP-PromptUser in Section 4.3.6.1. In the textbox of the window that opens as a result, which is shown in Figure 54, enter **user3**, the name of the database account recreated in Test LiUA-ListAccounts in Section 4.3.5.3.

Expected Results: The feature should prompt the user to enter the new password to be assigned to the database account.

Test Results: The window shown in Figure 56 opens.



**Figure 56. Window Prompting for the New Password to be Assigned to the Database Account**

Test Status: Passed.

#### 4.3.6.4 Test CUP-FailToAssignUserPassword

Goal: Verify that the feature terminates if the user fails within three attempts to enter and confirm the new password to be assigned to the account.

Test Description: In the currently open window, which is shown in Figure 56, enter, three times in succession, information *other than* the same valid password in the **Password** field and the **Verify** field, clicking **OK** after doing so each time. For example, you may:

- leave both fields, or just the **Verify** field, empty;
- enter different passwords in both fields;
- enter the same short password (less than 6 characters) in both fields; or
- enter the same long password (more than 14 characters) in both fields.

In each case, a message window opens to explain your error. (This behavior is programmed into the COEPromptPassword tool, not into the ELIST Database Utility Segment.) After the first

two failures, click **OK** to close that window and reopen the window shown in Figure 56. After the third failure, click **OK** again.

Expected Results: The feature should inform the user of the failure and then terminate.

Test Results: The window shown in Figure 57 opens.



**Figure 57. Window Confirming Failure to Assign a New Password to the Database Account**

Click **OK**.

Test Status: Passed.

#### **4.3.6.5 Test CUP-PromptSystemPassword**

Goal: Verify that, after successfully entering and confirming the new password to be assigned to the account, the user is prompted to enter the `SYSTEM` password.

Test Description: Repeat Test CUP-PromptUserPassword in Section 4.3.6.3. In the window that opens as a result, which is shown in Figure 56, enter the same valid password in the **Password** field and the **Verify** field (*i.e.*, a password between 6 and 14 characters in length) and click **OK**.

Expected Results: The feature should prompt the user to enter the password assigned to the `SYSTEM` user of the ELIST database instance.

Test Results: The window shown in Figure 31 opens.

Test Status: Passed.

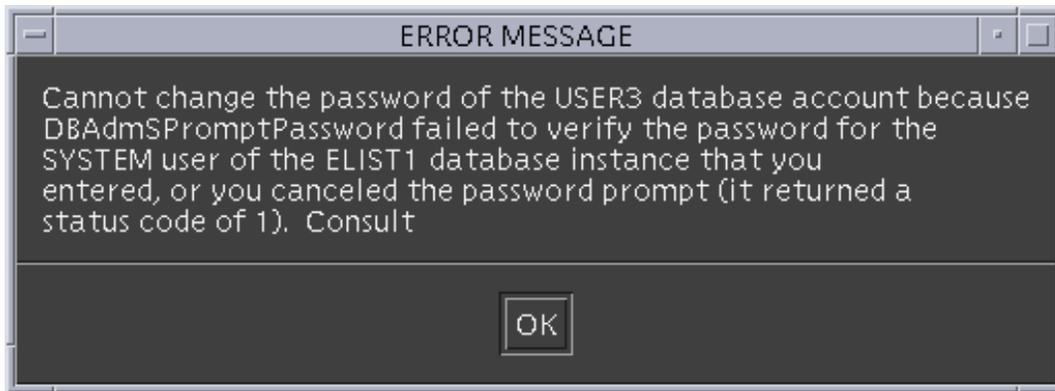
#### **4.3.6.6 Test CUP-FailToEnterSystemPassword**

Goal: Verify that the feature terminates if the user fails within three attempts to enter the `SYSTEM` password.

Test Description: In the currently open window, which is shown in Figure 31, enter, three times in succession, an incorrect password, clicking **OK** after doing so each time. A message window opens to explain your error. (This behavior is programmed into the `DBAdmSPromptPassword` API of the `DBAdmS` segment, not into the ELIST Database Utility Segment.) After the first two failures, click **OK** to close that window and reopen the window shown in Figure 31. After the third failure, click **OK** again.

Expected Results: The feature should inform the user of the failure and terminate.

Test Results: The window shown in Figure 58 opens.



**Figure 58. Window Confirming Failure to Enter the **SYSTEM** Password of the **ELIST** Database Instance**

Click **OK**.

Test Status: Passed.

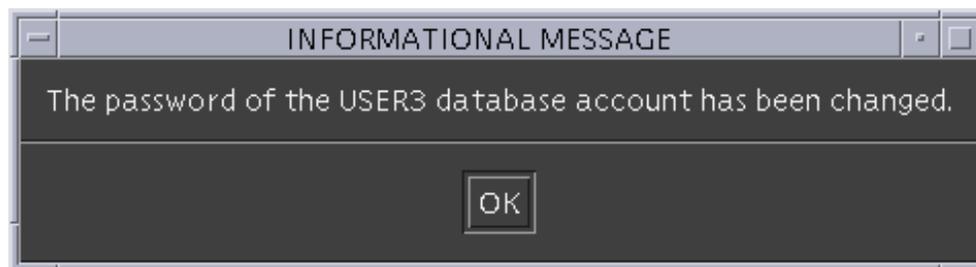
#### **4.3.6.7 Test CUP-ChangePassword**

Goal: Verify that, after successfully entering the **SYSTEM** password, the user is informed that the password of the database account has been changed, assuming the account exists.

Test Description: Repeat Test CUP-PromptSystemPassword in Section 4.3.6.5. In the window that opens as a result, which is shown in Figure 31, enter the correct password of the **SYSTEM** user of the **ELIST** database instance and click **OK**.

Expected Results: In a minute or so, the feature should display a message confirming that it has changed the password of the database account.

Test Results: The window shown in Figure 59 opens.



**Figure 59. Window Confirming that the Password of the Database Account Has Been Changed**

Click **OK**.

Test Status: Passed.

#### **4.3.6.8 Test CUP-CheckAccount**

Goal: Verify that the password of the database account has been changed.

Test Description: Log out as an administrative ELIST user and log back in as a DBA (see the “Account Preparation” section of the *ELIST IP*). Launch the **SQL Plus** feature of the **ORAS** application. When a message window prompting for the “Oracle TWO\_TASK” opens, enter the name of the ELIST database instance and click **OK**. A terminal window opens next. In response to the prompt for a user-name, enter the name supplied in Test CUP-PromptUserPassword in Section 4.3.6.3 (*i.e.*, **user3**), followed by a carriage return. In response to the prompt for that user’s password, enter the password assigned in Test CUP-PromptSystemPassword in Section 4.3.6.5, followed by a carriage return.

Expected Results: SQL\*Plus should accept the user-name and corresponding password, connect to Oracle, and prompt for an SQL command with SQL>.

Test Results: The terminal window opened by SQL\*Plus appears as shown in Figure 34 at this point.

Enter **exit**, followed by a carriage return. Log out as a DBA and log back in as an administrative ELIST user.

Test Status: Passed.

#### 4.3.6.9 Test CUA-Nonexistent

Goal: Verify that, after successfully entering the **SYSTEM** password, the user is informed and the feature terminates if the database account does not exist.

Test Description: Remove the database account for **user3** by repeating Test RUA-RemoveAccount in Section 4.3.2.5. Disregarding the fact that **user3** no longer exists, repeat Test CUP-PromptSystemPassword in Section 4.3.6.5. In the window that opens as a result, which is shown in Figure 31, enter the correct password of the **SYSTEM** user of the ELIST database instance and click **OK**.

Expected Results: The feature should display a message indicating that account **user3** does not exist.

Test Results: The window shown in Figure 41 opens.

Click **OK**.

Test Status: Passed.

## 4.4 Functional Tests of the Administrative Features of the ELIST Software Segment

The tests in this section demonstrate that administrative ELIST users can launch the **Add Map Data** and **Delete Map Data** features of the ELIST Software Segment on both the database server and a separate application client.

### 4.4.1 Add Map Data Feature

To test this feature, you must have a NIMA map CD in raster product format. Any CD containing ADRG, CADRG, or DTED data will suffice.

You should be currently logged in on the database server as an administrative ELIST user. The tests will be performed there first, then they will be repeated on the application client.

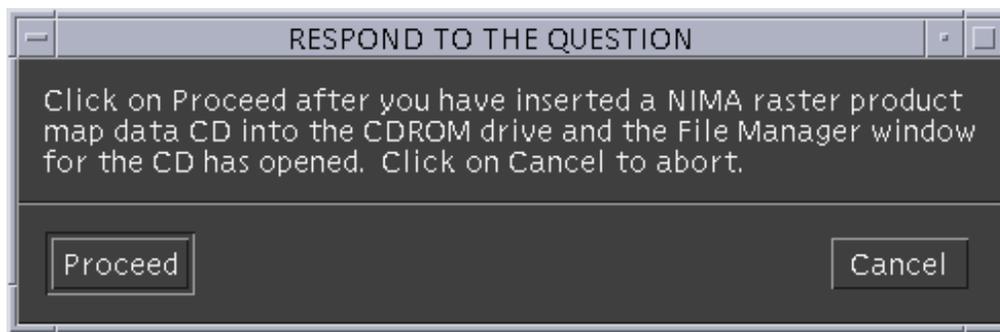
#### 4.4.1.1 Test AMD-PromptUser

Goal: Verify that the user is prompted to insert a CD and then choose between **Proceed** and **Cancel**.

Test Description: Launch the **Add Map Data** feature of the ELIST Software Segment as directed in the *User's Manual (UM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Software Segment*.

Expected Results: The feature should display a message telling the user how to proceed.

Test Results: The window shown in Figure 60 opens.



**Figure 60. Add Map Data Window with Instructions for Proceeding**

Test Status: Passed.

#### 4.4.1.2 Test AMD-NoCDLoaded

Goal: Verify that the prompt is repeated if the user clicks **Proceed** before inserting a CD.

Test Description: In the currently open window, which is shown in Figure 60, click **Proceed** before (i.e., without) inserting a CD.

Expected Results: The feature should repeat the prompt.

Test Results: The window shown in Figure 60 closes and reopens.

Test Status: Passed.

#### 4.4.1.3 Test AMD-CancelAdd

Goal: Verify that the feature terminates if **Cancel** is clicked.

Test Description: In the currently open window, which is shown in Figure 60, click **Cancel**.

Expected Results: The feature should display a message indicating that it is aborting.

Test Results: The window shown in Figure 61 opens.



**Figure 61. Add Map Data Window after Clicking Cancel**

Click **OK**.

Test Status: Passed.

#### **4.4.1.4 Test AMD-PrematureResponse**

Goal: Verify that the prompt is repeated if the user clicks **Proceed** after inserting a CD but before the File Manager window for the CD opens.

Test Description: Repeat Test AMD-PromptUser in Section 4.4.1.1. The window that opens as a result is shown in Figure 60. *Before* clicking **Proceed**, insert into the CDROM drive a NIMA map CD for data that has not already been added to the ELIST Reference Data Segment, and for which space is available on the volume that contains that segment. After doing so, but before the File Manager window for the CD opens, click **Proceed** (it is necessary to work quickly to accomplish that).

Expected Results: The feature should repeat the prompt.

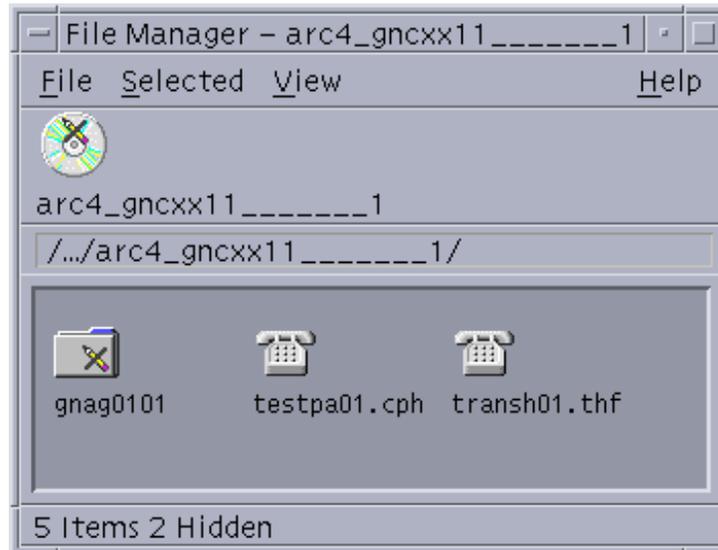
Test Results: The window shown in Figure 60 closes and reopens.

Test Status: Passed.

#### **4.4.1.5 Test AMD-TimelyResponse**

Goal: Verify that the prompt is *not* repeated if the user clicks **Proceed** after the File Manager window for the CD opens.

Test Description: Wait until the File Manager window for the CD opens. When it does, it will look like the window shown in Figure 62. (For these tests, a CD of ADRG data named arc4\_gncxx11\_\_\_\_\_1 was used.)



**Figure 62. File Manager Window that Opens after a NIMA Map CD is Inserted into the CDROM Drive**

After this window opens, click **Proceed** in the window shown in Figure 60.

Expected Results: The window shown in Figure 60 should *not* be redisplayed. Instead, a long delay should begin, during which the map data are copied from the CD to the ELIST Reference Data Segment.

Test Results: The window shown in Figure 60 closes but does not reopen. A long delay begins, during which the CDE performance meter shows heavy I/O activity.

Test Status: Passed.

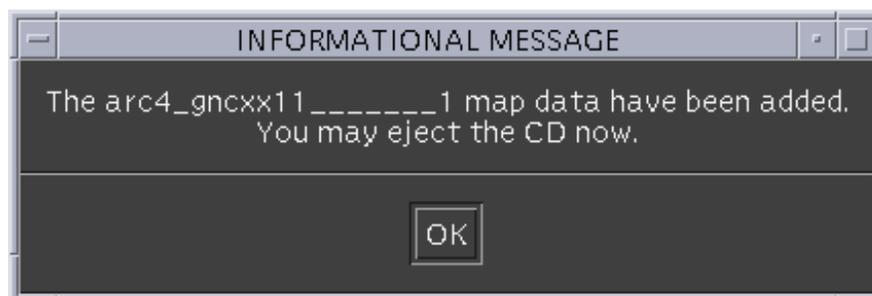
#### 4.4.1.6 Test AMD-AddNew

Goal: Verify that, if the map data set on the CD is not already present in the ELIST Reference Data Segment, it is added if space is available.

Test Description: Continue to wait while the processing started in the previous test continues and eventually ends.

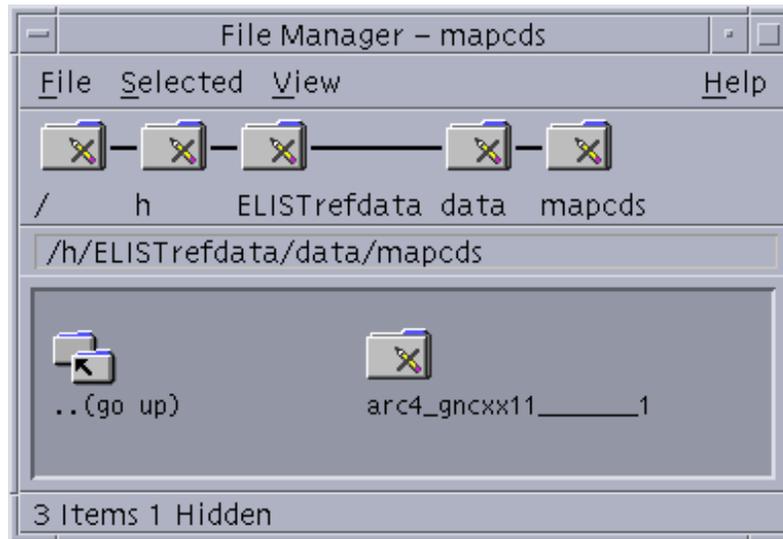
Expected Results: The feature should confirm that the map data have been added to the ELIST Reference Data Segment.

Test Results: After a suitable delay, the window shown in Figure 63 opens.



**Figure 63. Add Map Data Window Confirming Adding of Map Data**

Click **OK** (but do not eject the CD yet). Further confirmation is obtained by opening a File Manager window and navigating to the /h/ELISTrefdata/data/mapcds directory, which contains a subdirectory called arc4\_gncxx1\_\_\_\_1, as shown in Figure 64.



**Figure 64. File Manager Window Showing the Added Map Data**

Test Status: Passed.

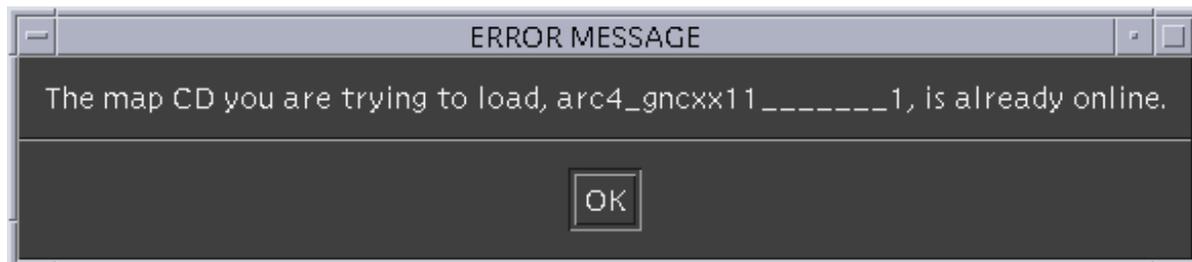
#### 4.4.1.7 Test AMD-AddDuplicate

Goal: Verify that, if the map data set on the CD *is* already present in the ELIST Reference Data Segment, it is *not* added.

Test Description: With the CD you just used still inserted in the CDROM drive, launch the **Add Map Data** feature again. The window that opens as a result is shown in Figure 60. Click **Proceed**.

Expected Results: The feature should display a message indicating that the map data you are trying to add are already present in the ELIST Reference Data Segment (*i.e.*, have previously been added).

Test Results: The window shown in Figure 65 opens.



**Figure 65. Add Map Data Window Indicating that Map Data are Already Online**

Click **OK**.

Test Status: Passed.

#### 4.4.1.8 Test AMD-NoSpace

Goal: Verify that, if the map data set on the CD is not already present in the ELIST Reference Data Segment, it is *not* added if space is unavailable.

Test Description: This test can only be performed if insufficient space remains on the volume containing the ELIST Reference Data Segment to copy the map data from the CD to that segment. One way to simulate these conditions is to open a terminal window, `su` to `root`, and copy a sufficiently large file or files onto that volume, if necessary, before beginning the test. (The subdirectory just added to `/h/ELISTrefdata/data/mapcds`, *i.e.* `arc4_gncxx11_____1`, may conveniently be recursively copied one or more times to `/h/ELISTrefdata/data/mapcds` or elsewhere on the volume, giving it a new name each time.) As additional preparation for this test, and while still connected as `root`, also rename the subdirectory just added (*i.e.*, `arc4_gncxx11_____1`) as something else, allowing another copy of it to be added. Then, with the CD you just used still inserted in the CDROM drive, launch the **Add Map Data** feature again. The window that opens as a result is shown in Figure 60. Click **Proceed**.

Expected Results: Assuming insufficient space remains to add the map data again, the feature should inform the user of that fact and not add the data.

Test Results: The window shown in Figure 66 opens.<sup>10</sup>



**Figure 66. Add Map Data Window Indicating Insufficient Space**

Click **OK** and eject the CD. Be sure to reverse the temporary steps taken to ensure insufficient space and to rename the previously added map data.

Test Status: Passed.

#### 4.4.1.9 Test AMD-Client

Goal: Verify that the feature works correctly on a separate application client.

Test Description: Log out on the database server, and log back in as an administrative ELIST user on a separate application client. Repeat Test AMD-PromptUser in Section 4.4.1.1 through

<sup>10</sup> Apparently, the text of the message is too long for the tool (COEInstError) used to display the message and is truncated. The last sentence of the message should read as follows: "Use the 'Delete Map Data' feature to delete some data to make more space available."

Test AMD-NoSpace in Section 4.4.1.8. When finished, log out on the application client, and log back in as an administrative ELIST user on the database server.

Expected Results: The results previously obtained on the database server should be obtained on the application client.

Test Results: Though not shown here, the same results are obtained.

Test Status: Passed.

#### 4.4.2 Delete Map Data Feature

You should be currently logged in on the database server as an administrative ELIST user. The tests will be performed there first, then they will be repeated on the application client.

##### 4.4.2.1 Test DMD-ListDataSets

Goal: Verify that, if one or more map data sets are present in the ELIST Reference Data Segment, the user is presented with a list of their names and is prompted to enter one of them.

Test Description: This test can only be performed if one or more map data sets are present in the ELIST Reference Data Segment; that condition holds, of course, after the previous test. Launch the **Delete Map Data** feature of the ELIST Software Segment as directed in the *User's Manual (UM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Software Segment*.

Expected Results: The **Delete Map Data** feature should display a message listing the map data sets currently online in the ELIST Reference Data Segment and telling the user how to proceed.

Test Results: The window shown in Figure 67 opens.

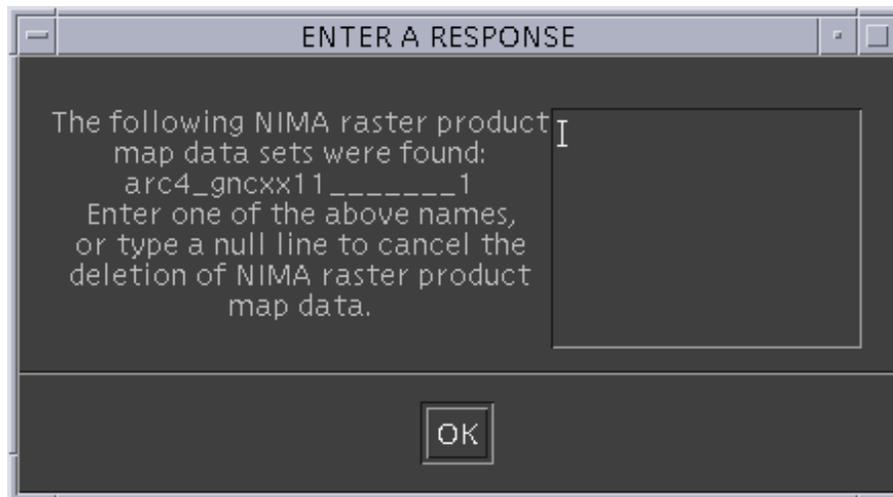


Figure 67. Delete Map Data Window Showing the Online Map Data Sets

Test Status: Passed.

#### 4.4.2.2 Test DMD-CancelDelete

Goal: Verify that the feature terminates if the user clicks **OK** without entering the name of a map data set.

Test Description: In the currently open window, which is shown in Figure 67, leave the textbox empty and click **OK**.

Expected Results: The **Delete Map Data** feature should display a message indicating that it is aborting.

Test Results: The window shown in Figure 68 opens.



**Figure 68. Delete Map Data Window after Selecting No Data to Delete**

Click **OK**.

Test Status: Passed.

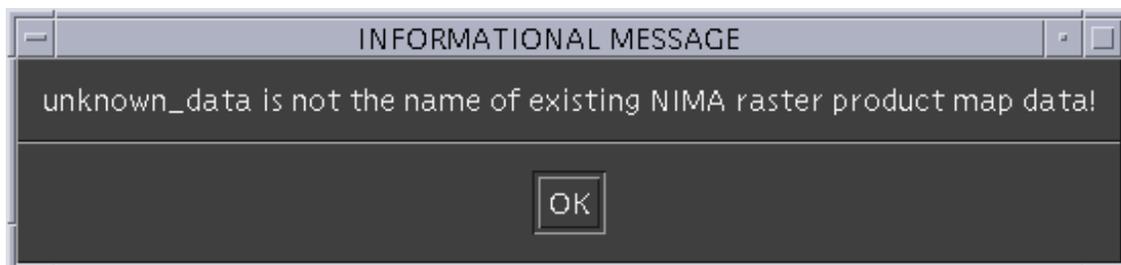
#### 4.4.2.3 Test DMD-Nonexistent

Goal: Verify that, if the name entered is *not* actually that of an existing map data set, the user is informed and is given another chance.

Test Description: Relaunch the **Delete Map Data** feature. The window that opens as a result is shown in Figure 67. In the window's textbox, enter `unknown_data` (or any other name *not* in the list displayed in the window) and click **OK**.

Expected Results: The **Delete Map Data** feature should inform the user that the named data is not online and should repeat the previous prompt.

Test Results: The window shown in Figure 69 opens.



**Figure 69. Delete Map Data Window after Specifying Nonexistent Data to Delete**

Click **OK**. The window shown in Figure 67 reopens.

Test Status: Passed.

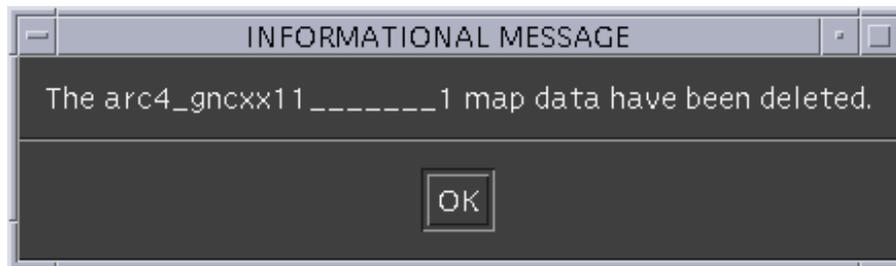
#### 4.4.2.4 Test DMD-Delete

Goal: Verify that, if the user enters the name of an existing map data set, the map data set is deleted.

Test Description: In the currently open window, which is shown in Figure 67, enter arc4\_gncxx11 \_\_\_\_\_ 1 (or any other name in the list displayed in the window) and click **OK**.

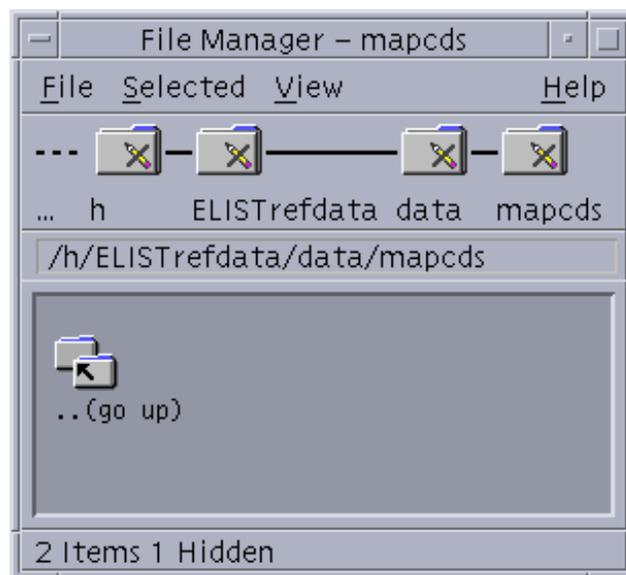
Expected Results: The feature should confirm that the map data have been deleted from the ELIST Reference Data Segment

Test Results: After a suitable delay, the window shown in Figure 70 opens.



**Figure 70. Delete Map Data Window after Deleting Map Data**

Click **OK**. Further confirmation is obtained by opening a File Manager window and navigating to the /h/ELISTrefdata/data/mapcdfs directory, which no longer contains a subdirectory with the name of the data that were just deleted, as shown in Figure 71.



**Figure 71. File Manager Window Confirming the Deletion of Map Data**

Test Status: Passed.

#### 4.4.2.5 Test DMD-NoData

Goal: Verify that, if no map data sets are present in the ELIST Reference Data Segment, the user is so informed.

Test Description: This test can only be performed if no map data sets are present in the ELIST Reference Data Segment; that condition holds, of course, after the previous test. Relaunch the **Delete Map Data** feature.

Expected Results: The feature should inform the user that there is no map data to delete.

Test Results: The window shown in Figure 72 opens.



**Figure 72. Delete Map Data Window when No Map Data Sets Exist**

Click **OK**.

Test Status: Passed.

#### 4.4.2.6 Test DMD-Client

Goal: Verify that the feature works correctly on a separate application client.

Test Description: Log out on the database server, and log back in as an administrative ELIST user on a separate application client. Repeat Test DMD-ListDataSets in Section 4.4.2.1 through Test DMD-NoData in Section 4.4.2.5. When finished, log out on the application client, and log back in as an administrative ELIST user on the database server.

Expected Results: The results previously obtained on the database server should be obtained on the application client.

Test Results: Though not shown here, the same results are obtained.

Test Status: Passed.

### 4.5 Functional Tests of the General Features of the ELIST Software Segment

The tests in this section demonstrate that general ELIST users can launch the **Run ELIST** and **Run ETEdit** features of the ELIST Software Segment on both the database server and a separate application client.

Log in to the database server platform as a general ELIST user who has been given an account in the ELIST database instance (see Section 4.3.1). Follow the instructions in this section for launching the features.

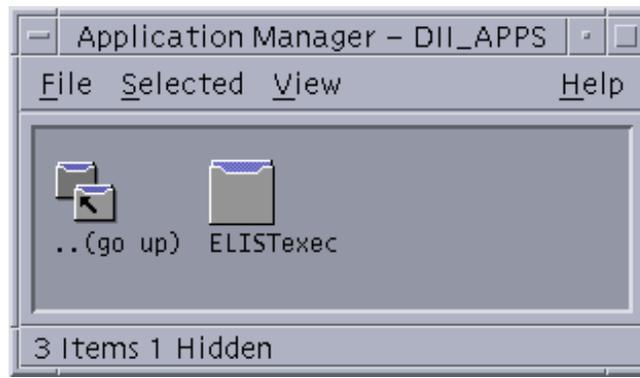
#### 4.5.1 Run ELIST Feature

This feature is tested by launching it from its desktop icon.

##### 4.5.1.1 Test REI-Launch

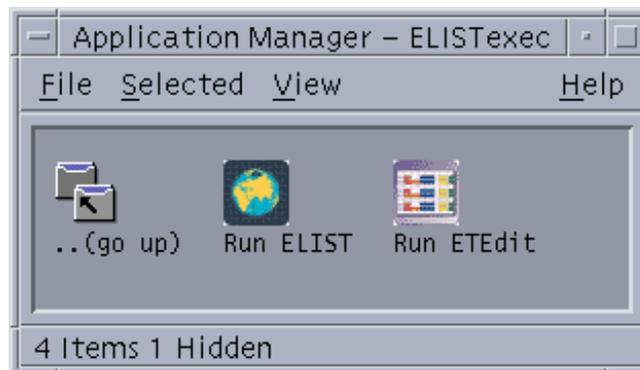
Goal: Verify that general ELIST users can launch ELIST.

Test Description: Using the right mouse button on the desktop, select **Applications** → **Application Manager**. The Application Manager window opens. Double-click **DII\_APPS**. The Application Manager – DII\_APPS window opens, as shown in Figure 73.



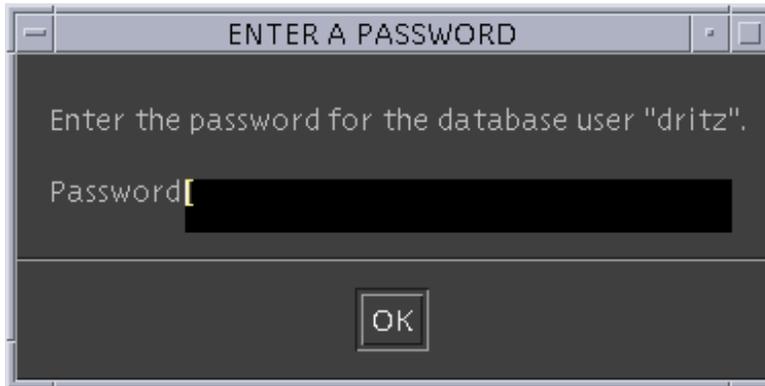
**Figure 73. Application Manager – DII\_APPS Window**

Using the left mouse button, double-click **ELISTExec**. The Application Manager – ELISTExec window opens, as shown in Figure 74.



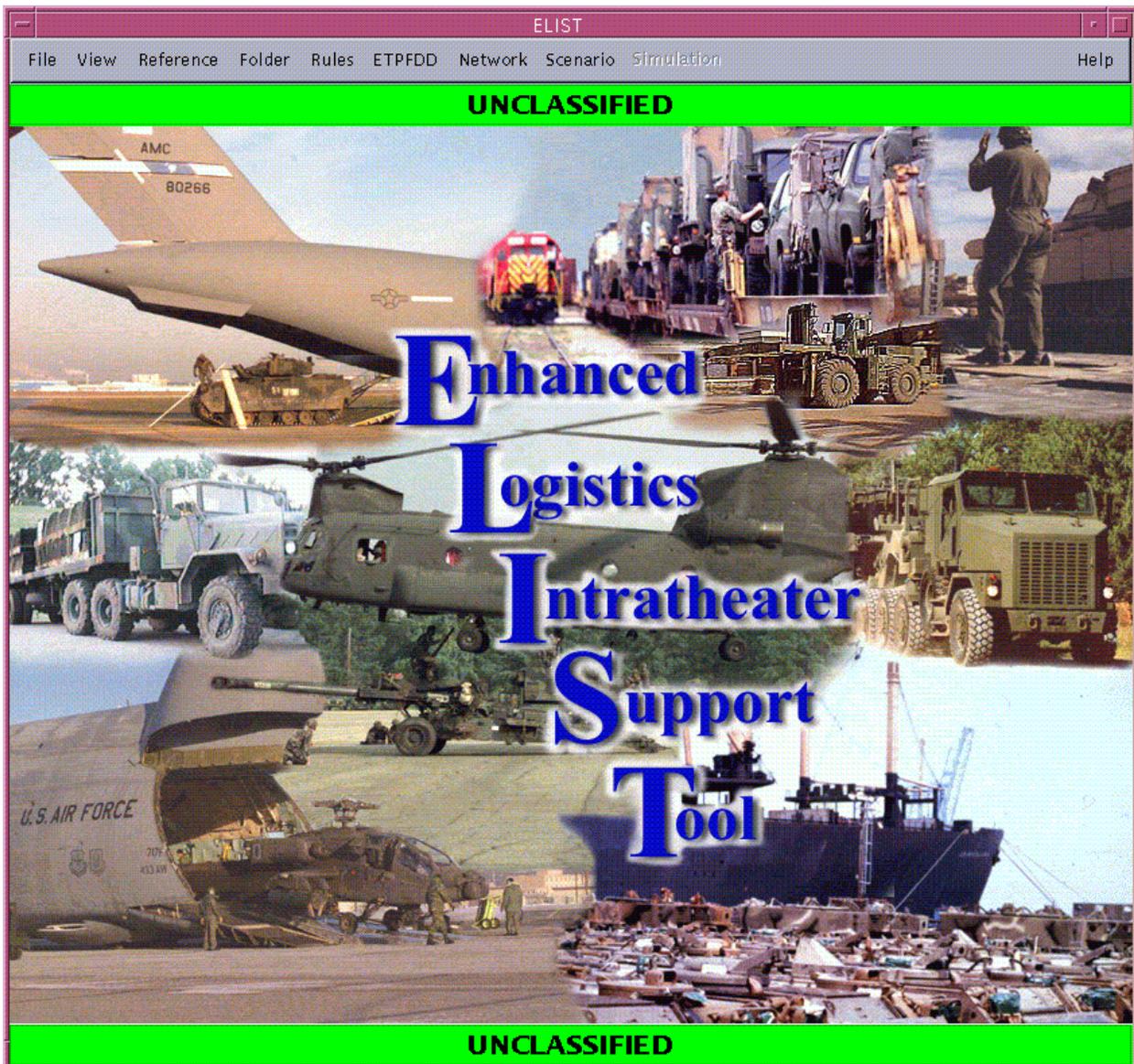
**Figure 74. Application Manager – ELISTExec Window**

Using the left mouse button, double-click **Run ELIST**. In a few seconds, a terminal window opens. (This window contains progress and diagnostic messages from ELIST.) Move this window out of the way (for example, to the lower right-hand corner of the desktop). A moment later, ELIST prompts for the user's database account password, as shown in Figure 75.



**Figure 75. Prompt for the User’s Database Password**

Enter the password into the textbox and click **OK**. After a delay, during which progress messages are scrolled to the terminal window, the ELIST logo window opens, as shown in Figure 76.

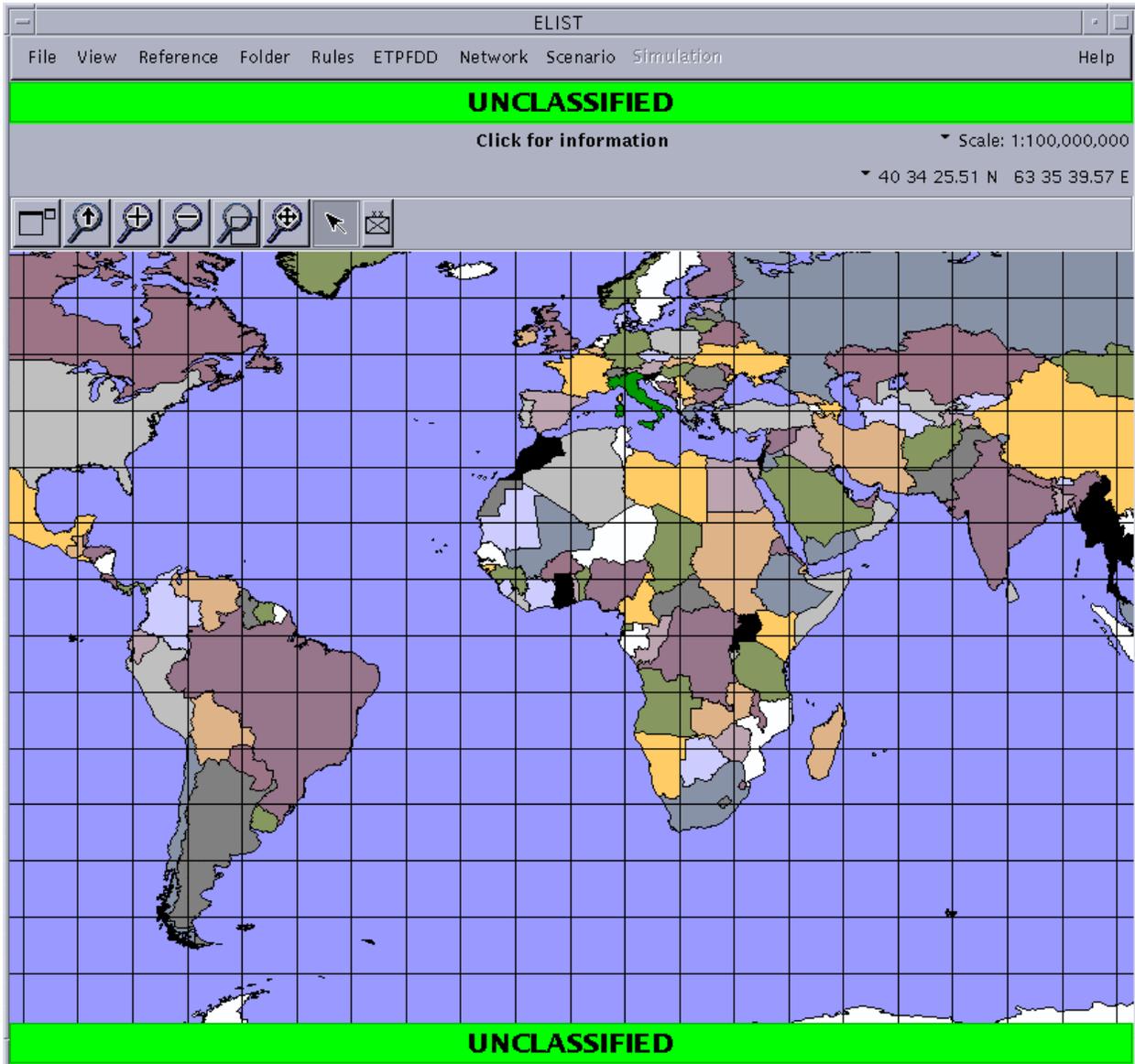


**Figure 76. ELIST Logo Window**

Initialization continues, during which tables are loaded from the ELIST database and further progress messages are scrolled to the terminal window.

Expected Results: The ELIST logo window should be replaced by the ELIST map window.

Test Results: After a long delay, the window shown in Figure 76 is replaced by the window shown in Figure 77.



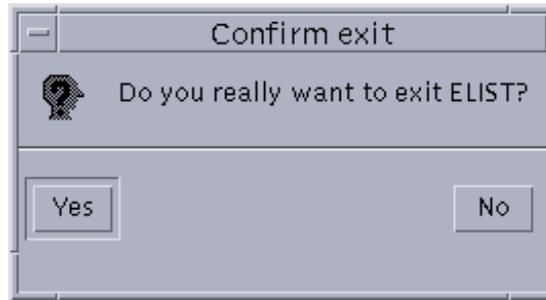
**Figure 77. ELIST Map Window**

Test Status: Passed.

#### 4.5.1.2 Test REI-Terminate

Goal: Verify that the feature can be terminated.

Test Description: In the currently open window, which is shown in Figure 77, select **File** → **Exit**. As shown in Figure 78, a confirmation window opens.



**Figure 78. Confirm Termination of ELIST Window**

Click **Yes**.

Expected Results: The feature should terminate. Before doing so, it should pause to give the operator a chance to copy any messages of interest in the terminal window.

Test Results: The ELIST map window closes and the window shown in Figure 79 opens.



**Figure 79. Window Affording an Opportunity to Pause Before Closing the ELIST Terminal Window**

Click **OK**. The terminal window closes.

Test Status: Passed.

#### **4.5.1.3 Test REI-Client**

Goal: Verify that the feature works correctly on a separate application client.

Test Description: Log out on the database server, and log back in again as a general ELIST user on a separate application client. Repeat Test REI-Launch in Section 4.5.1.1 and Test REI-Terminate in Section 4.5.1.2. When finished, log out on the application client, and log back in as a general ELIST user on the database server.

Expected Results: The results previously obtained on the database server should be obtained on the application client.

Test Results: Though not shown here, the same results are obtained.

Test Status: Passed.

#### **4.5.2 Run ETEdit Feature**

This feature is tested by launching it from its desktop icon.

#### 4.5.2.1 Test REt-Launch

Goal: Verify that general ELIST users can launch ETEdit.

Test Description: Using the right mouse button on the desktop, select **Applications** → **Application Manager**. The Application Manager window opens. Double-click **DII\_APPS**. The Application Manager – DII\_APPS window opens, as shown in Figure 73.

Using the left mouse button, double-click **ELISTexec**. The Application Manager – ELISTexec window opens, as shown in Figure 74.

Using the left mouse button, double-click **Run ETEdit**. In a few seconds, a terminal window opens. (This window contains progress and diagnostic messages from ETEdit.) Move this window out of the way (for example, to the lower right-hand corner of the desktop). A moment later, ETEdit prompts for the user's database account password, as shown in Figure 75.

Enter the password into the textbox and click **OK**.

Expected Results: After a delay, during which progress messages are scrolled to the terminal window, the ETEdit logo window opens, as shown in Figure 80.



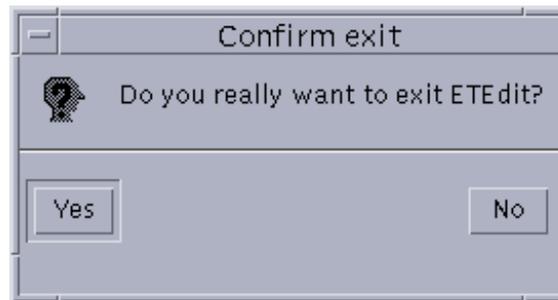
Figure 80. ETEdit Logo Window

Test Status: Passed.

#### 4.5.2.2 Test REt-Terminate

Goal: Verify that the feature can be terminated.

Test Description: In the currently open window, which is shown in Figure 80, select **File** → **Exit**. As shown in Figure 81, a confirmation window opens.

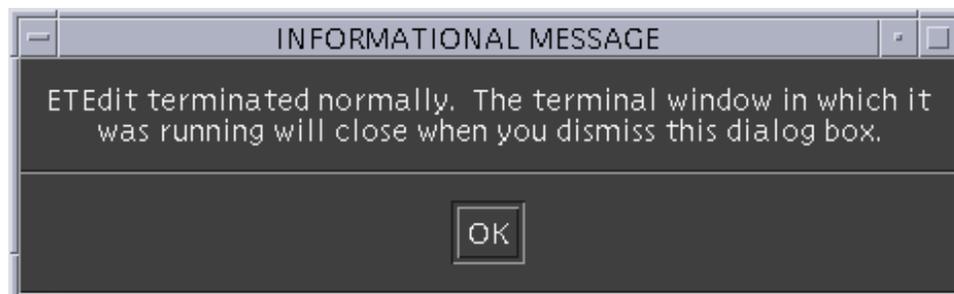


**Figure 81. Confirm Termination of ETEdit Window**

Click **Yes**.

Expected Results: The feature should terminate. Before doing so, it should pause to give the operator a chance to copy any messages of interest in the terminal window.

Test Results: The ETEdit logo window closes and the window shown in Figure 82 opens.



**Figure 82. Window Affording an Opportunity to Pause Before Closing the ETEdit Terminal Window**

Click **OK**. The terminal window closes.

Test Status: Passed.

#### 4.5.2.3 Test REt-Client

Goal: Verify that the feature works correctly on a separate application client.

Test Description: Log out on the database server, and log back in again as a general ELIST user on a separate application client. Repeat Test REt-Launch in Section 4.5.2.1 and Test REt-Terminate in Section 4.5.2.2. When finished, log out on the application client.

Expected Results: The results previously obtained on the database server should be obtained on the application client.

Test Results: Though not shown here, the same results are obtained.

Test Status: Passed.

*(Additional tests are still to be provided.)*

## 4.6 Deinstallation Tests and Results

The tests in this section demonstrate that all the segments of the ELIST mission application deinstall correctly. In these tests, the segments are deinstalled in the proper order, which is the reverse of their installation order.

Log in to the database server platform as the system administrator (`sysadmin` user). Follow the instructions in this section for deinstalling the segments. The instructions here make liberal reference to the procedures for deinstalling particular segments as presented in the *ELIST IP*; consult the *ELIST IP* when directed to do so.

### 4.6.1 ELIST Reference Data Segment

Because it is not interactive, demonstrating that the deinstallation of the ELIST Reference Data Segment is performed correctly requires only one test on the database server, which is then repeated on the application client.

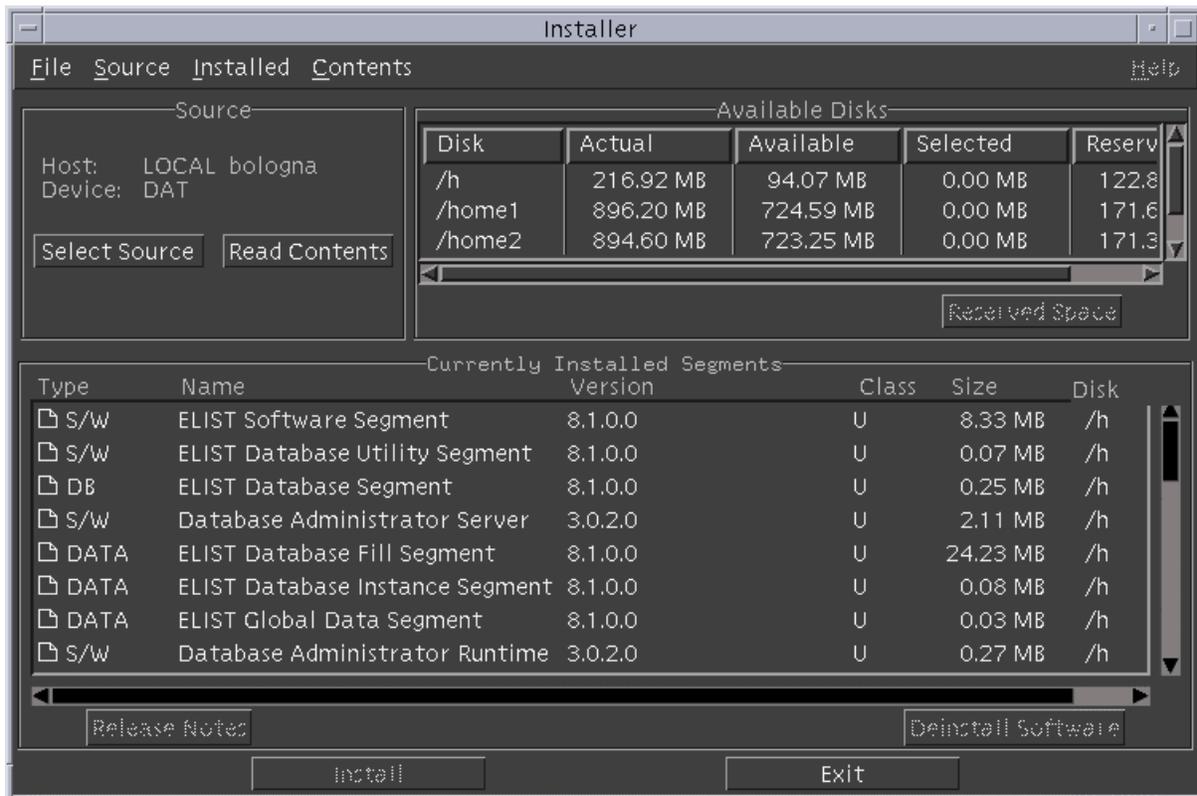
#### 4.6.1.1 Test DRDS-FinishServerDeinstall

Goal: Verify that the ELIST Reference Data Segment deinstalls correctly on the database server.

Test Description: Follow the steps in the *ELIST IP* for deinstalling the ELIST Reference Data Segment.

Expected Results: The Segment Installer should complete the deinstallation of the ELIST Reference Data Segment.

Test Results: As shown in Figure 83, the Segment Installer window reopens, with the ELIST Reference Data Segment absent from the list of currently installed segments.



**Figure 83. Segment Installer Window after Deinstalling the ELIST Reference Data Segment on the Database Server**

Test Status: Passed.

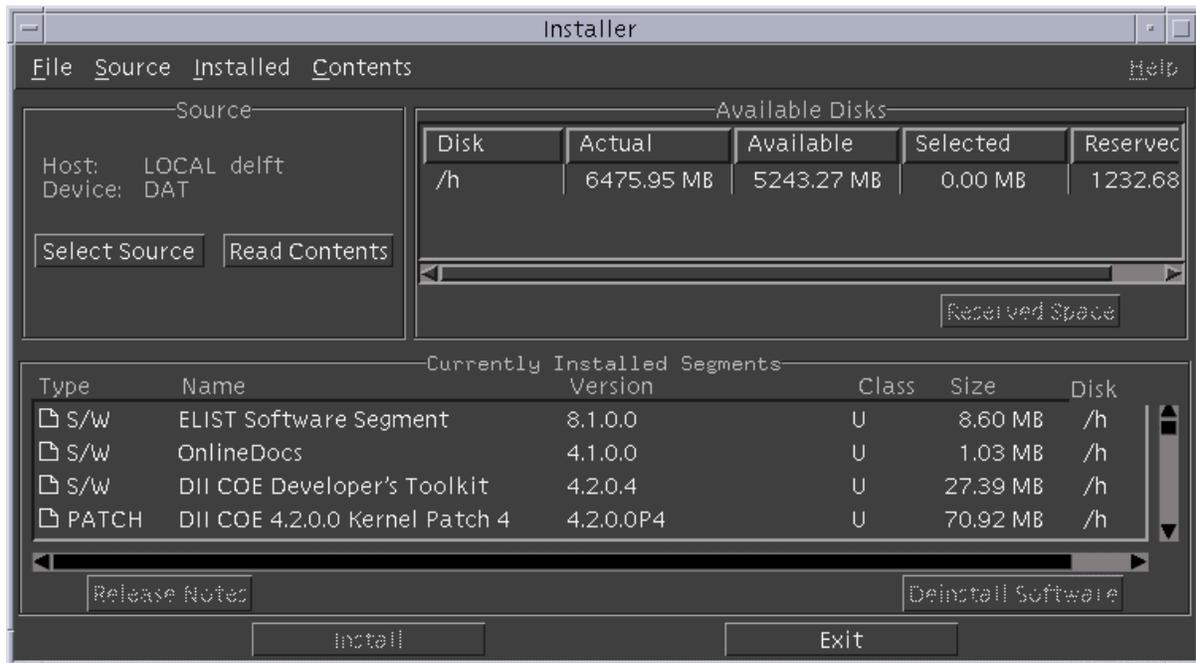
#### 4.6.1.2 Test DRDS-FinishClientDeinstall

Goal: Verify that the ELIST Reference Data Segment deinstalls correctly on a separate application client.

Test Description: Log out on the database server and log back in as `sysadmin` on a separate application client. Follow the steps in the *ELIST IP* for deinstalling the ELIST Reference Data Segment.

Expected Results: The Segment Installer should complete the deinstallation of the ELIST Reference Data Segment.

Test Results: As shown in Figure 84, the Segment Installer window reopens, with the ELIST Reference Data Segment absent from the list of currently installed segments.



**Figure 84. Segment Installer Window after Deinstalling the ELIST Reference Data Segment on the Application Client**

Test Status: Passed.

## 4.6.2 ELIST Software Segment

Because it is not interactive, demonstrating that the deinstallation of the ELIST Server Segment is performed correctly requires only one test on the database server, which is then repeated on the application client.

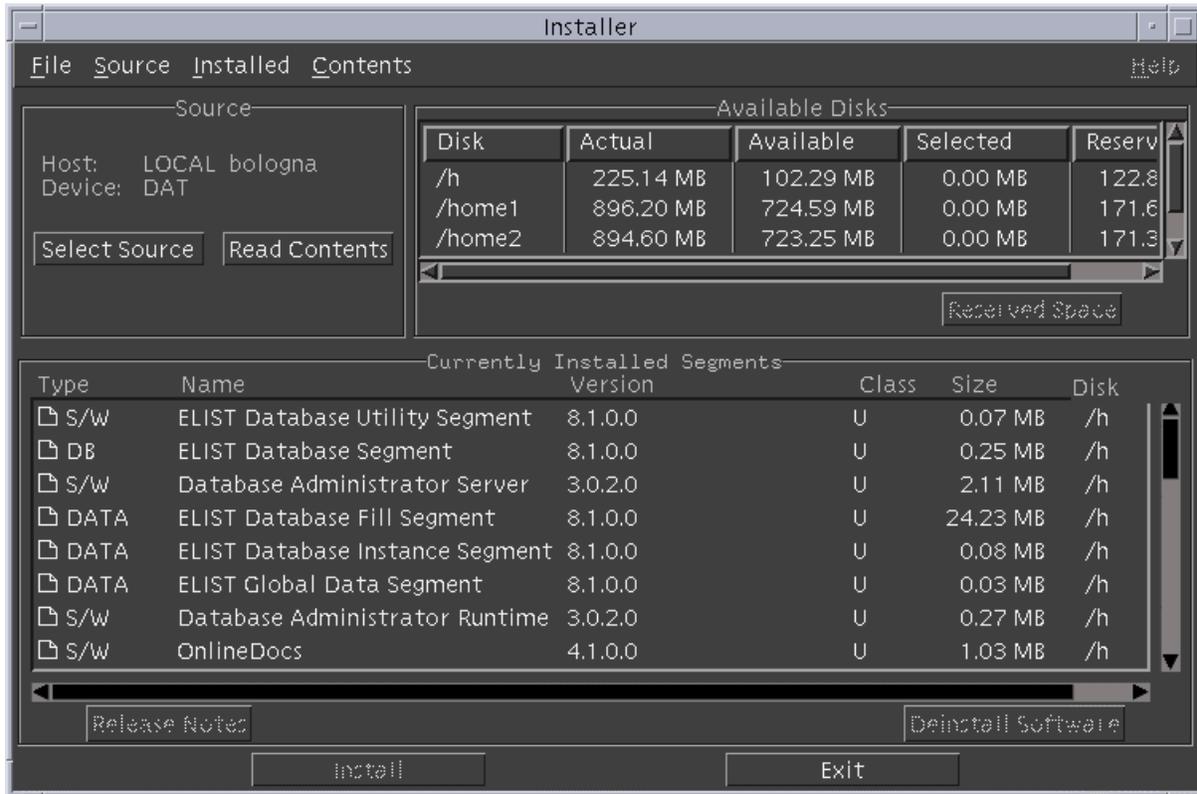
### 4.6.2.1 Test DSS-FinishServerDeinstall

Goal: Verify that the ELIST Software Segment deinstalls correctly on the database server.

Test Description: Log out on the application client and log back in as `sysadmin` on the database server. Follow the steps in the *ELIST IP* for deinstalling the ELIST Software Segment.

Expected Results: The Segment Installer should complete the deinstallation of the ELIST Software Segment.

Test Results: As shown in Figure 85, the Segment Installer window reopens, with the ELIST Software Segment absent from the list of currently installed segments.



**Figure 85. Segment Installer Window after Deinstalling the ELIST Software Segment on the Database Server**

Test Status: Passed.

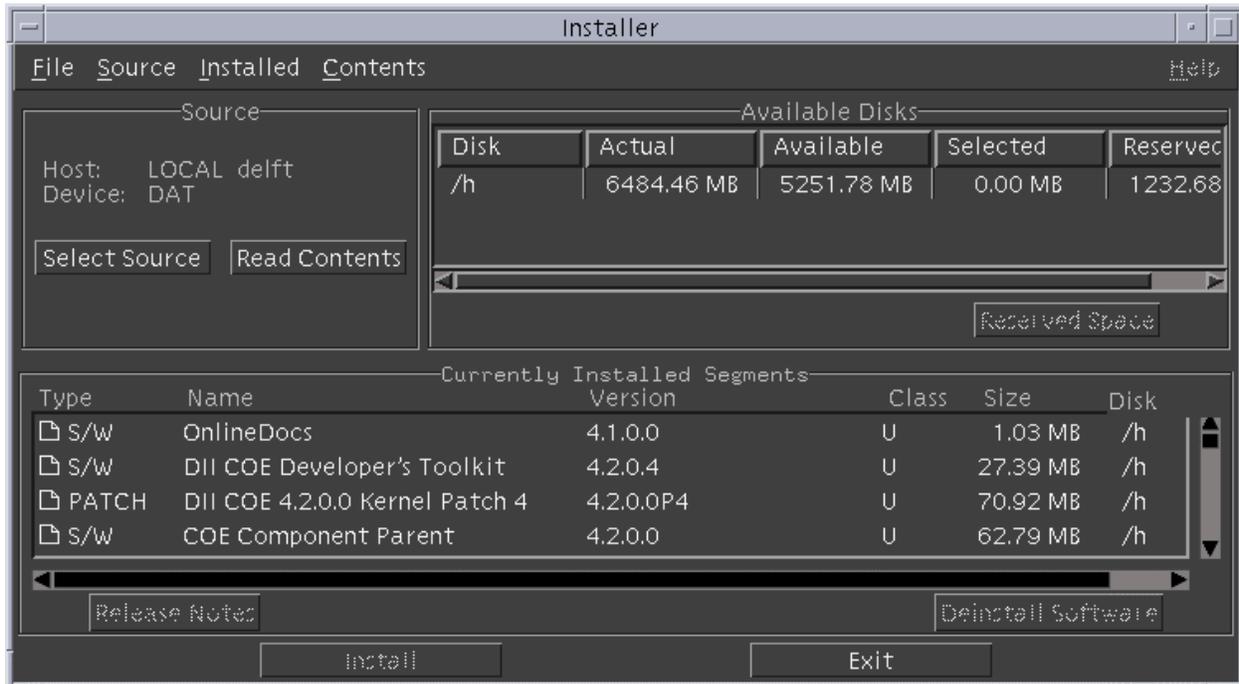
#### 4.6.2.2 Test DSS-FinishClientDeinstall

Goal: Verify that the ELIST Software Segment deinstalls correctly on a separate application client.

Test Description: Log out on the database server and log back in as `sysadmin` on a separate application client. Follow the steps in the *ELIST IP* for deinstalling the ELIST Software Segment.

Expected Results: The Segment Installer should complete the deinstallation of the ELIST Software Segment.

Test Results: As shown in Figure 86, the Segment Installer window reopens, with the ELIST Software Segment absent from the list of currently installed segments.



**Figure 86. Segment Installer Window after Deinstalling the ELIST Software Segment on the Application Client**

Test Status: Passed.

### 4.6.3 ELIST Database Utility Segment

Because it is not interactive, demonstrating that the deinstallation of the ELIST Database Utility Segment is performed correctly requires only one test.

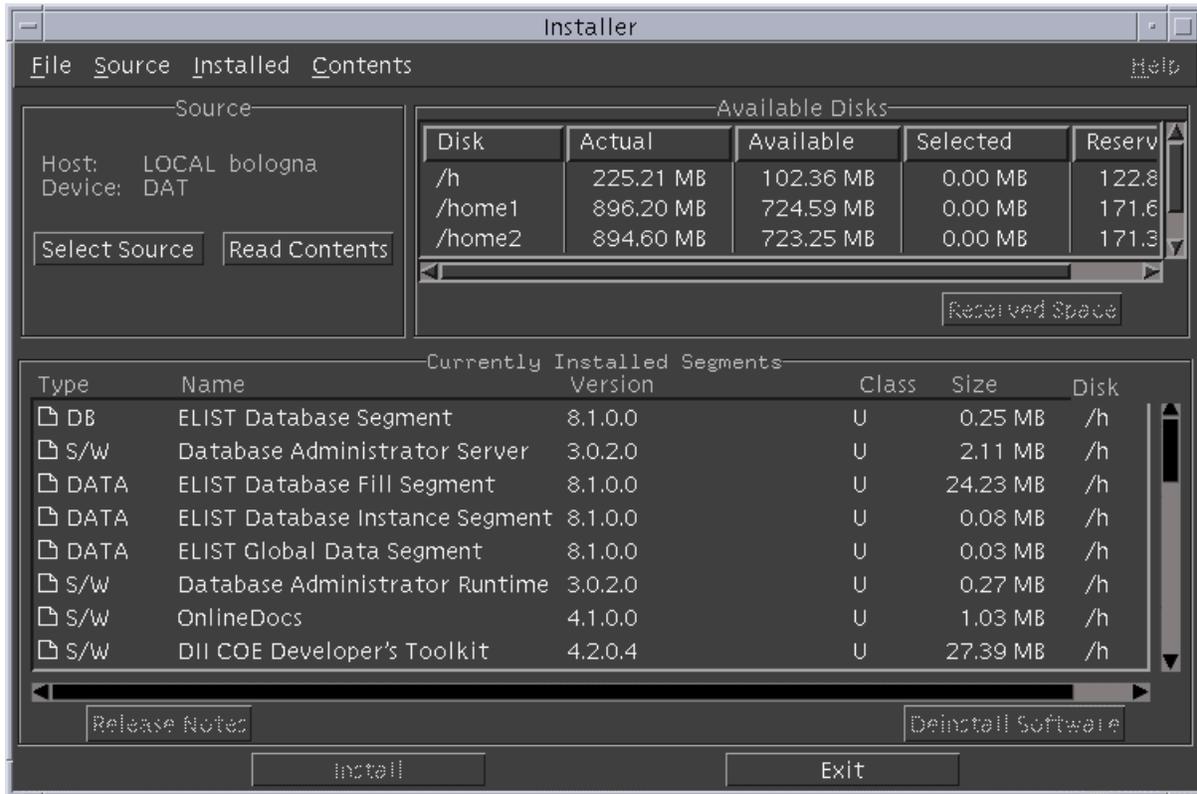
#### 4.6.3.1 Test DDUS-FinishServerDeinstall

Goal: Verify that the ELIST Database Utility Segment deinstalls correctly.

Test Description: Log out on the application client and log back in as `sysadmin` on the database server. Follow the steps in the *ELIST IP* for deinstalling the ELIST Database Utility Segment.

Expected Results: The Segment Installer should complete the deinstallation of the ELIST Database Utility Segment.

Test Results: As shown in Figure 87, the Segment Installer window reopens, with the ELIST Database Utility Segment absent from the list of currently installed segments.



**Figure 87. Segment Installer Window after Deinstalling the ELIST Database Utility Segment**

Test Status: Passed.

#### 4.6.4 ELIST Database Segment

Because it is interactive, demonstrating that the installation of the ELIST Database Segment is performed correctly requires a sequence of small tests.

##### 4.6.4.1 Test DDS-PromptForSystemPassword

Goal: Verify that the administrator is prompted to enter the previously assigned password of the SYSTEM user of the ELIST database instance.

Test Description: Follow the steps in the *ELIST IP* for deinstalling the ELIST Database Segment, stopping when prompted to enter the SYSTEM password.

Expected Results: The Segment Installer should prompt for the password previously assigned to the SYSTEM user of the ELIST database instance (the instance called **ELIST1** here).

Test Results: The window shown in Figure 18 opens.

Test Status: Passed.

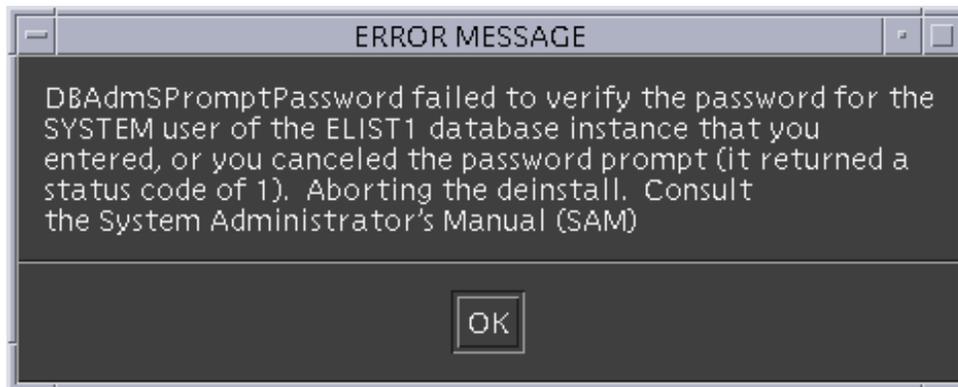
##### 4.6.4.2 Test DDS-CancelSystemPasswordPrompt

Goal: Verify that the deinstallation is aborted if the administrator clicks **Cancel** instead of entering the SYSTEM password.

Test Description: In the currently open window, which is shown in Figure 18, leave the textbox empty and click **Cancel**.

Expected Results: The Segment Installer should abort the deinstallation of the ELIST Database Segment.

Test Results: The window shown in Figure 88 opens.<sup>11</sup>



**Figure 88. Message Noting Failure to Satisfy Prompt for SYSTEM Password**

Test Status: Passed.

#### 4.6.4.3 Test DDS-FailToEnterSystemPassword

Goal: Verify that the deinstallation is aborted if, after three attempts, the administrator fails to enter the SYSTEM password successfully.

Test Description: Repeat Test DDS-PromptForSystemPassword in Section 4.6.4.1. In the window that opens as a result, which is shown in Figure 18, enter, three times in succession, an incorrect password, clicking **OK** after doing so each time. A message window opens to explain your error. (This behavior is programmed into the DBAdmSPromptPassword API of the DBAdmS segment, not into the ELIST Database Segment.) After the first two failures, click **OK** to close that window and reopen the window shown in Figure 18. After the third failure, click **OK** again.

Expected Results: The Segment Installer should abort the deinstallation of the ELIST Database Segment.

Test Results: The window shown Figure 88 opens.

Upon clicking **OK** in that window, the Segment Installer window shown in Figure 87 reopens. (The ELIST Database Segment remains in the list of currently installed segments.)

Test Status: Passed.

<sup>11</sup> Apparently, the text of the message is too long for the tool (COEInstError) used to display the message and is truncated. The last sentence of the message should read as follows: "Consult the System Administrator's Manual (SAM) for this segment."

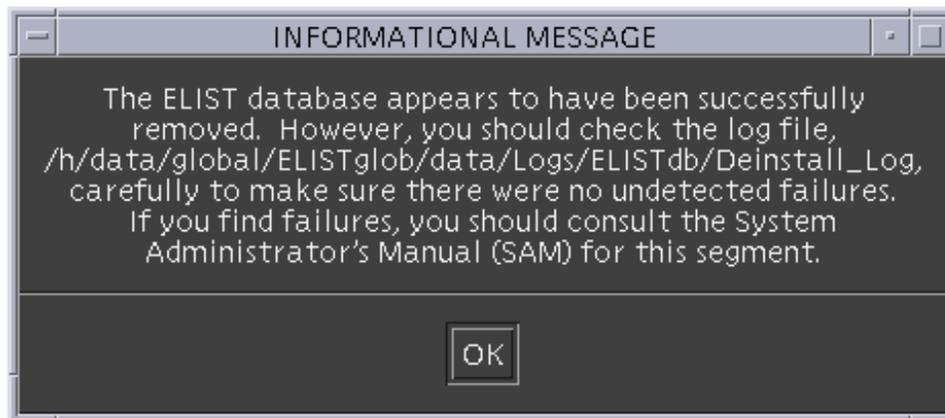
#### 4.6.4.4 Test DDS-RemoveDatabase

Goal: Verify that, after successfully entering the `SYSTEM` password, the administrator is informed that the database has been removed.

Test Description: Repeat Test DDS-PromptForSystemPassword in Section 4.6.4.1. In the window that opens as a result, which is shown in Figure 18, enter the correct password of the `SYSTEM` user of the ELIST database instance and click **OK**.

Expected Results: After a delay, the Segment Installer should inform the administrator that the ELIST database has been successfully removed.

Test Results: The window shown in Figure 89 opens.



**Figure 89. Message Noting Successful Removal of the ELIST Database**

Test Status: Passed.

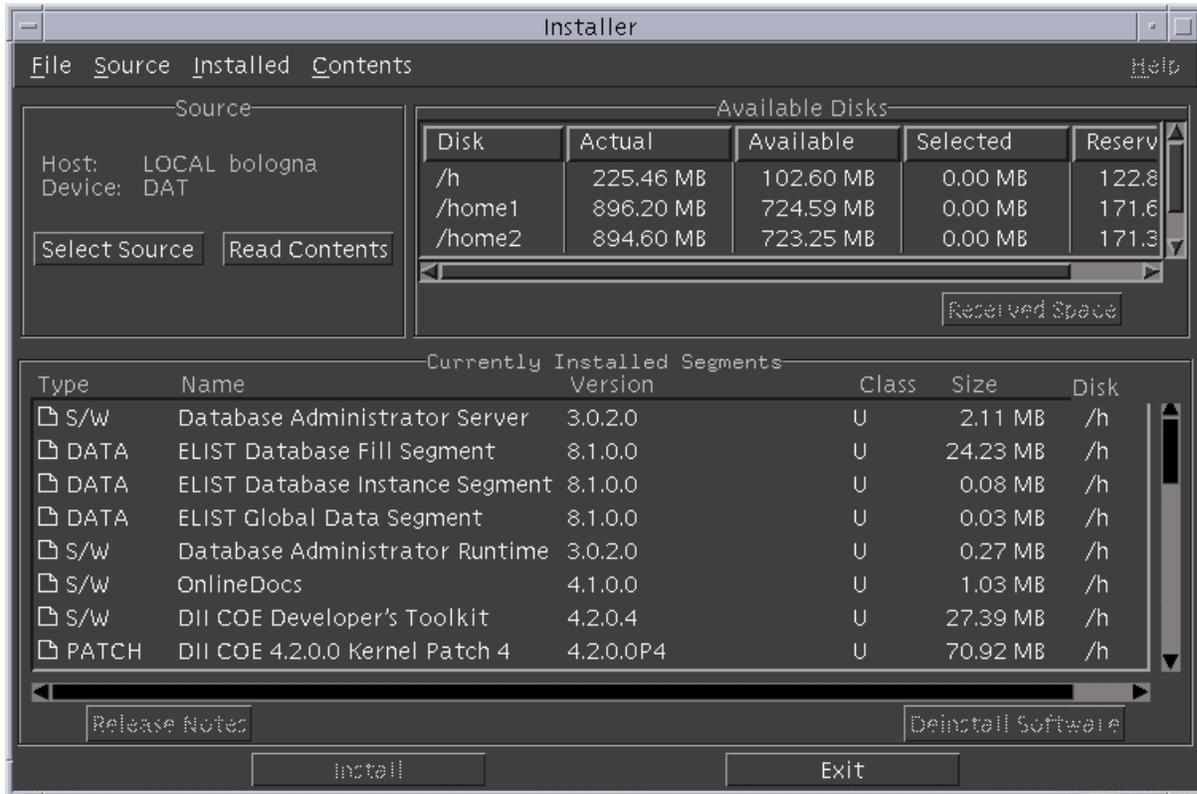
#### 4.6.4.5 Test DDS-FinishDeinstall

Goal: Verify that the ELIST Database Segment deinstalls correctly.

Test Description: In the currently open window, which is shown in Figure 89, click **OK**.

Expected Results: The Segment Installer should complete the deinstallation of the ELIST Database Segment

Test Results: As shown in Figure 90, the Segment Installer window reopens, with the ELIST Database Segment absent from the list of currently installed segments.



**Figure 90. Segment Installer Window after Deinstalling the ELIST Database Segment**

Test Status: Passed.

#### 4.6.5 ELIST Database Fill Segment

Because it is not interactive, demonstrating that the deinstallation of the ELIST Database Fill Segment is performed correctly requires only one test.

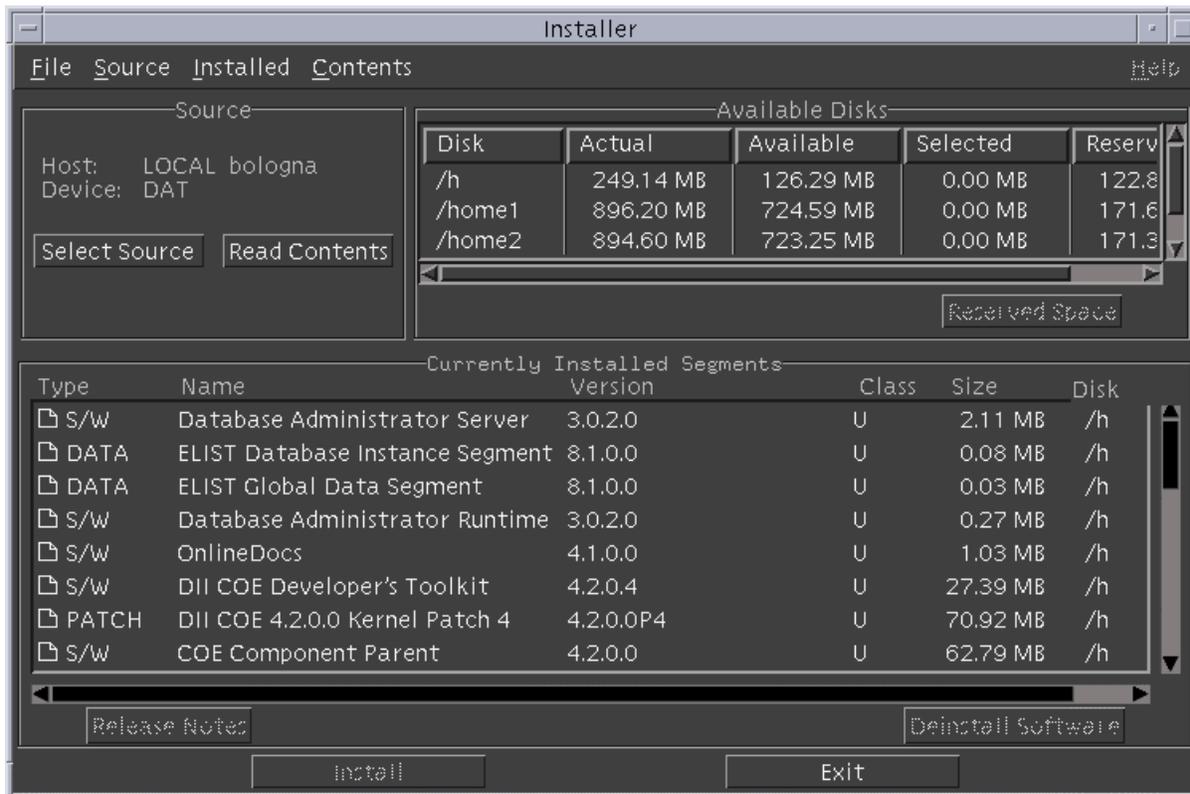
##### 4.6.5.1 Test DDIS-FinishDeinstall

Goal: Verify that the ELIST Database Fill Segment deinstalls correctly.

Test Description: Follow the instructions in the *ELIST IP* for deinstalling the ELIST Database Fill Segment.

Expected Results: The Segment Installer should complete the deinstallation of the ELIST Database Fill Segment.

Test Results: As shown in Figure 91, the Segment Installer window reopens, with the ELIST Database Fill Segment absent from the list of currently installed segments.



**Figure 91. Segment Installer Window after Deinstalling the ELIST Database Fill Segment**

Test Status: Passed.

#### 4.6.6 ELIST Database Instance Segment

Because it is interactive, demonstrating that the deinstallation of the ELIST Database Instance Segment is performed correctly requires a sequence of small tests.

##### 4.6.6.1 Test DDIS-RemoveNewInstance

Goal: Verify that, if **Create new instance** was chosen when the segment was installed, the administrator is informed that the database instance has been removed.

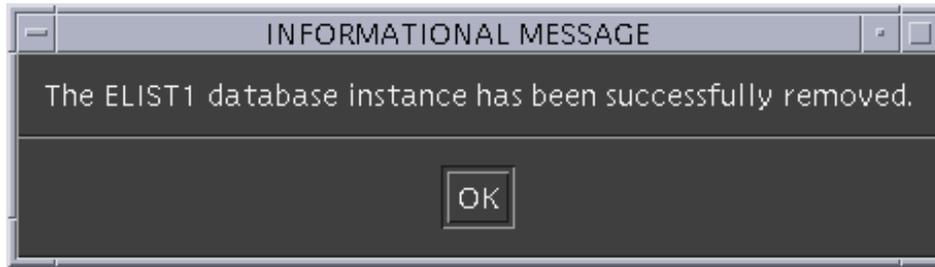
Test Description: This test can only be performed if **Create new instance** was chosen when the segment was installed.

**NOTE:** If **Use existing instance** was chosen, proceed to Test DDIS-RemoveExistingInstance in Section 4.6.6.3.

Follow the instructions in the *ELIST IP* for deinstalling the ELIST Database Instance Segment, stopping when a message noting the removal of the database instance is displayed.

Expected Results: The Segment Installer should display a message informing the administrator that the ELIST database instance has been removed.

Test Results: The window shown in Figure 92 opens. (For this test, the ELIST database instance was named ELIST1.)



**Figure 92. Message Noting Successful Removal of the ELIST Database Instance**

Test Status: Passed.

#### 4.6.6.2 Test DDIS-FinishDeinstall

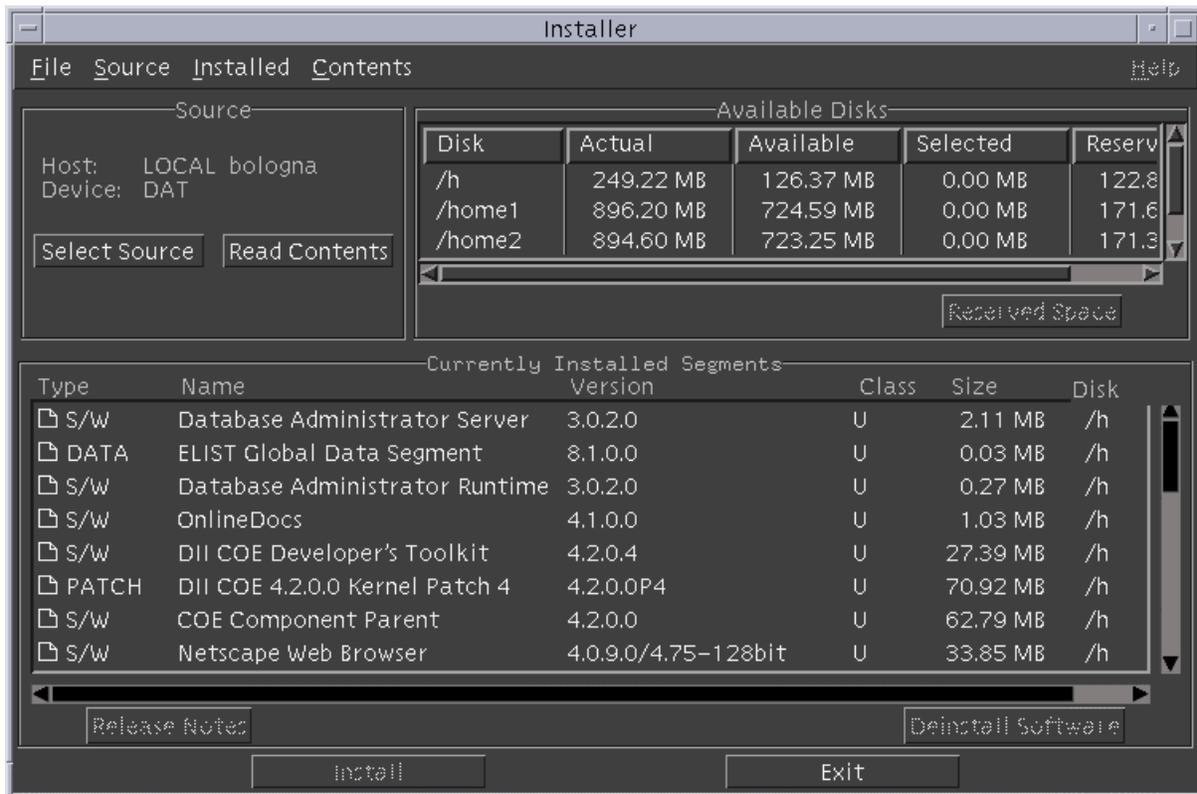
Goal: Verify that the ELIST Database Instance Segment deinstalls correctly when its installation created a new database instance.

Test Description: In the currently open window, which is shown in Figure 92, click **OK**.

Expected Results: The Segment Installer should complete the deinstallation of the ELIST Database Instance Segment, after which the following five directories and one file should no longer exist:

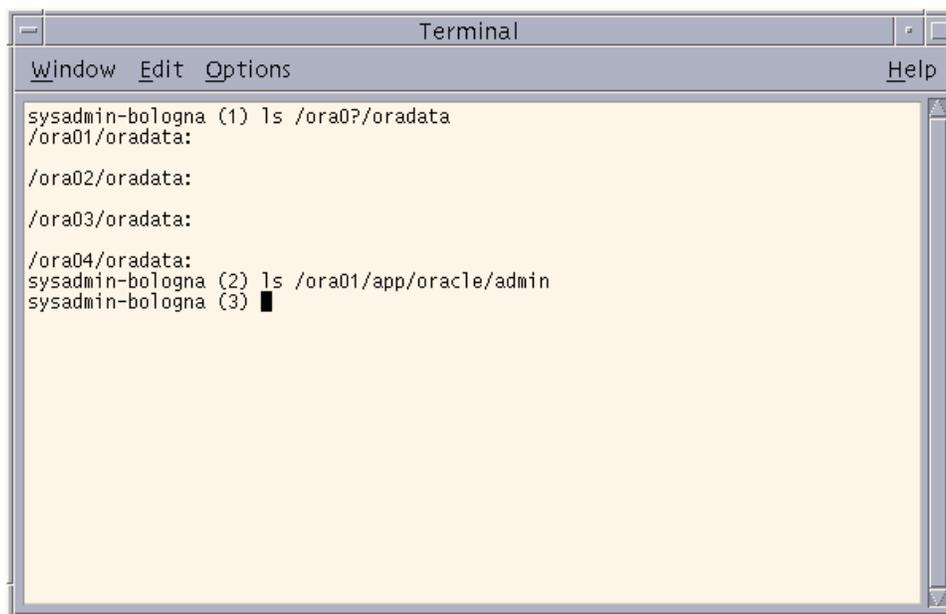
- /ora01/oradata/ELIST1
- /ora02/oradata/ELIST1
- /ora03/oradata/ELIST1
- /ora04/oradata/ELIST1
- /ora01/app/orace/admin/ELIST1
- /ora01/app/oracle/admin/ELIST1/ELIST1.cfg

Test Results: As shown in Figure 93, the Segment Installer window reopens, with the ELIST Database Instance Segment absent from the list of currently installed segments.



**Figure 93. Segment Installer Window after Deinstalling the ELIST Database Instance Segment**

Additionally, as shown Figure 94, the directories and file named above no longer exist.



**Figure 94. Terminal Window Showing that Instance Directories and Files Exist**

Test Status: Passed.

Next Test: Proceed to Test DGDS-FinishDeinstall in Section 4.6.7.1.

#### 4.6.6.3 Test DDIS-RemoveExistingInstance

Goal: Verify that, if **Use existing instance** was chosen when the segment was installed, the administrator is informed that the database instance has *not* been removed.

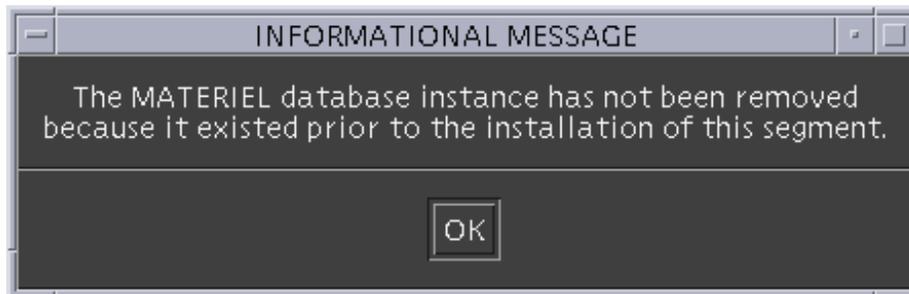
Test Description: This test can only be performed if **Create new instance** was chosen when the segment was installed.

**NOTE**: If **Use existing instance** was chosen, proceed to Test DDIS-RemoveNewInstance in Section 4.6.6.1 instead.

Follow the instructions in the *ELIST IP* for deinstalling the ELIST Database Instance Segment, stopping when a message noting the retention of the (preexisting) database instance is displayed.

Expected Results: The Segment Installer should display a message informing the administrator that the ELIST database instance has *not* been removed.

Test Results: The window shown in Figure 95 opens. (For this test, the name of the preexisting instance was MATERIEL.)



**Figure 95. Message Noting Retention of the Preexisting ELIST Database Instance**

Test Status: Passed.

#### 4.6.6.4 Test DDIS-FinishDeinstall2

Goal: Verify that the ELIST Database Instance Segment deinstalls correctly when its installation used an existing database instance.

Test Description: In the currently open window, which is shown in Figure 95, click **OK**.

Expected Results: The Segment Installer should complete the deinstallation of the ELIST Database Instance Segment.

Test Results: As shown in Figure 93, the Segment Installer window reopens, with the ELIST Database Instance Segment absent from the list of currently installed segments.

Test Status: Passed.

### 4.6.7 ELIST Global Data Segment

Because it is not interactive, demonstrating that the deinstallation of the ELIST Global Data Segment is performed correctly requires only one test.

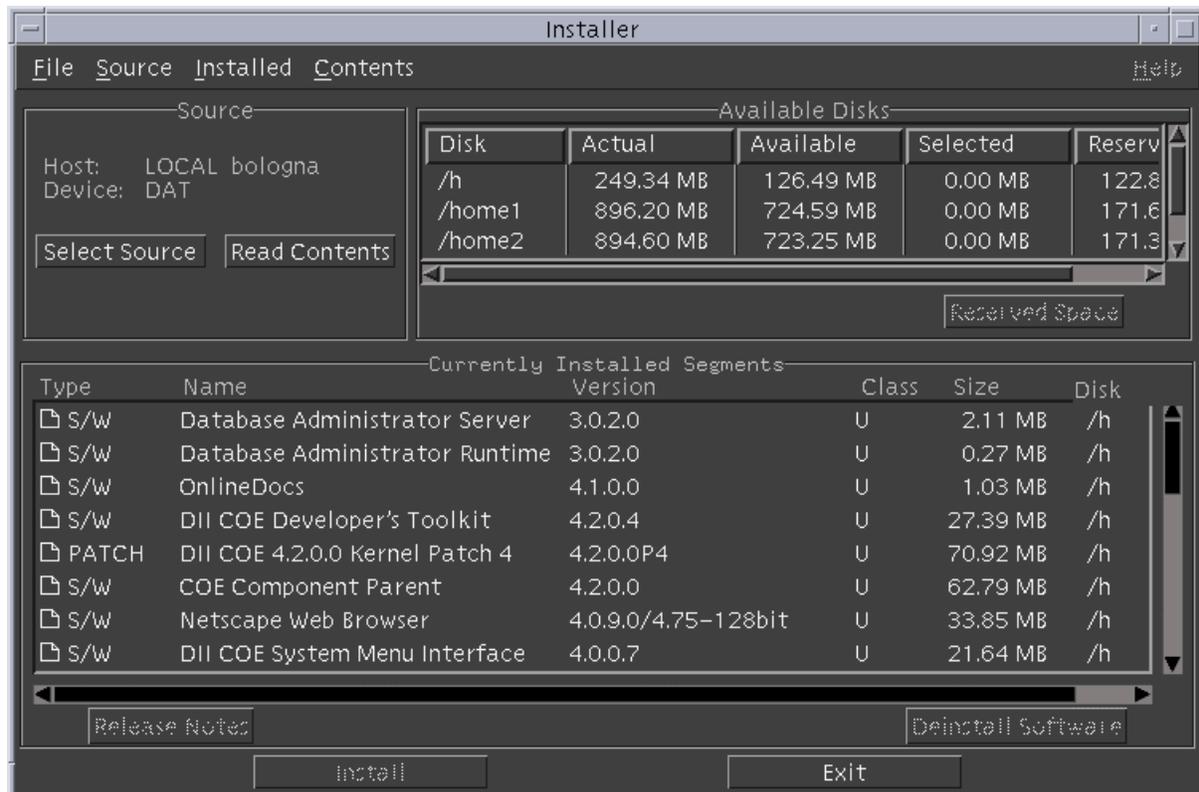
#### 4.6.7.1 Test DGDS-FinishDeinstall

Goal: Verify that the ELIST Global Data Segment deinstalls correctly.

Test Description: Follow the instructions in the *ELIST IP* for deinstalling the ELIST Global Data Segment.

Expected Results: The Segment Installer should complete the deinstallation of the ELIST Global Data Segment.

Test Results: As shown in Figure 96, the Segment Installer window reopens, with the ELIST Global Data Segment absent from the list of currently installed segments.



**Figure 96. Segment Installer Window after Deinstalling the ELIST Global Data Segment**

Test Status: Passed.

## 5. Notes

The following acronyms are (or may be) used in this document.

<b>Acronym</b>	<b>Definition</b>
ADRG	ARC Digitized Raster Graphics
CADRG	Compressed ARC Digitized Raster Graphics
CD	Compact Disk
CDE	Common Desktop Environment
CDROM	Compact Disk Read-Only Memory
CM	Configuration Management
COE	Common Operating Environment
COTS	Common Off-the-Shelf
CPU	Central Processor Unit
DB	Database
DBA	Database Administrator
DBDD	Database Design Document
DII	Defense Information Infrastructure
DII_APPS	DII Applications (desktop folder name)
DISA	Defense Information Systems Agency
DTED	Digital Terrain Elevation Data
ELIST	Enhanced Logistics Intratheater Support Tool (DII COE segment prefix)
ETEdit	ETPFDD Editor
ETPFDD	Enhanced Time-Phased Force Deployment Data
GUI	Graphical User Interface
IP	Installation Procedures
LAN	Local Area Network
MB	Megabytes
N/A	Not Applicable
NFS	Network File System
NIMA	National Imagery and Mapping Agency
NT	New Technology (an operating system for Microsoft Windows)
PC	Personal Computer
RDBMS	Relational Database Management System
SAM	System Administrator's Manual
SQL	Structured Query Language
SVD	Software Version Description
UM	User's Manual
WVS	World Vector Shoreline

This page intentionally left blank.

## **6. Acknowledgements**

Argonne National Laboratory is a Federally Funded Research and Development Center operated by The University of Chicago under contract W-31-109-ENG for the United States Department of Energy.

The development of ELIST and the preparation of this document were supported by funding from the Military Traffic Management Command Transportation Engineering Agency of the United States Army.

This page intentionally left blank

## **7. Documentation Improvement and Feedback**

Comments and other feedback on this document should be directed to:

Phone: (630) 252-7217

Fax: (630) 252-5128

Email: [dritz@anl.gov](mailto:dritz@anl.gov)

This page intentionally left blank.