

ELISTdbinst.8100.Final.SOL7.SAM

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

**System Administrator's Manual (SAM)
for the
Enhanced Logistics Intratheater Support Tool (ELIST)
Database Instance 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. Installation Overview	5
4. Error Recovery Guidelines.....	7
4.1 General Approach to Error Handling in the ELIST Database Instance Segment.....	7
4.1.1 General Approach to Error Handling During Installation	7
4.1.2 General Approach to Error Handling During Deinstallation.....	8
4.2 Handling of Specific Error Situations.....	9
4.2.1 Handling of Specific Errors During Installation.....	9
4.2.2 Handling of Specific Errors During Deinstallation	12
5. Creating an ELIST Database Instance Prior to Installing the ELIST Database Instance Segment.....	15
6. Changing the Password of the SYSTEM Account of the ELIST Database Instance.....	17
7. Notes.....	19
8. Acknowledgements	21
9. Documentation Improvement and Feedback.....	23

List of Tables

Table 1. Segments Comprising the ELIST Mission Application	1
--	---

1. Scope

This document is the *System Administrator's Manual (SAM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Instance Segment*. It covers errors that can arise during the segment's installation and deinstallation, and it outlines appropriate recovery actions. It also tells how to change the password for the SYSTEM account of the database instance after the instance is created, and it discusses the creation of a suitable database instance for ELIST by means other than the installation of the segment. The latter subject is covered in more depth than its introductory discussion in 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* (referred to in portions of this document as the *ELIST IP*).

The information in this document is expected to be of use only rarely. Other than errors arising from the failure to follow instructions, difficulties are not expected to be encountered during the installation or deinstallation of the segment. By the same token, the need to create a database instance for ELIST by means other than the installation of the segment is expected to be the exception, rather than the rule. Most administrators will only need to be aware of the help that is provided in this document and will probably not actually need to read and make use of it.

1.1 Identification

The ELIST Database Instance Segment is one of seven segments that make up the DII COE ELIST mission application.

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:

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

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

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.

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

² 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.

³To 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

Software Version Description (SVD) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Instance Segment Version 8.1.0.0 for Solaris 7, ELISTdbinst.8100.Final.SOL7.SVD, 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.8100.Final.SOL7.IP, Argonne National Laboratory, 26 February 2002

2.1.2 Other DII COE Documents

System Administrator's Manual (SAM) for the Database Administrator Runtime (DBAdmR) Version 3.0.2.0 for Solaris 7, CM Number 43773, FGM Inc., 4 April 2001

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 intentionally left blank.

3. Installation Overview

Installation and deinstallation of the ELIST Database Instance Segment are documented in the *ELIST IP*.

The *Software Version Description (SVD) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Instance Segment* alludes to the principal decision that must be made by the person installing the ELIST Database Instance Segment—namely, whether the segment should create a new database instance for ELIST or use an already existing one. The *ELIST IP* discusses the decision in further depth and covers the installation actions corresponding to each choice. It does not, however, describe how to create a database instance prior to the installation of the segment, if that is the path the installer chooses to take. Suggestions for accomplishing that goal are presented in this document (see Section 5).

This page intentionally left blank.

4. Error Recovery Guidelines

This section discusses the possible errors that can arise during the installation and deinstallation of the ELIST Database Instance Segment, and it presents guidance that can be helpful in recovering from those errors.

4.1 General Approach to Error Handling in the ELIST Database Instance Segment

4.1.1 General Approach to Error Handling During Installation

The ELIST Database Instance Segment is installed primarily for its side effect on the environment outside the segment, which is the creation of a new Oracle database instance for ELIST (or the association of an already existing instance with ELIST). In addition, information about the database instance is recorded in a data file of the (previously installed) ELIST Global Data Segment. That information includes, among other things,

- the name of the instance,
- the name of the database server platform (the machine on which the Oracle RDBMS is running and on which the database instance exists—which is also the machine on which the ELIST Global Data Segment and the ELIST Database Instance Segment are installed), and
- an indication of whether the installation of the ELIST Database Instance Segment created the instance (or, alternatively, whether it existed prior to the installation of the segment).

The principal issue influencing the approach to error handling during the installation of the ELIST Database Instance Segment is the realization that, even if something goes wrong part of the way through the process of creating the database instance, vestiges of that instance are likely to be left around in a partially created state. For example, the instance may have been “registered” with the ORACLE DataBase Instance (ORADBI) segment, altering the `data` directory of that segment. Similarly, administrative directories belonging to Oracle (typically in or underneath `/ora01`, `/ora02`, `/ora03`, and `/ora04`) may have been altered, and of course any of the various control files making up the instance may have been created and Oracle system tables updated. Recovery from this unpredictable error state requires that actions be taken either to make further forward progress towards the ultimate goal of creating the database instance or, alternatively, to reverse the external effects of its partial creation, restoring the state to that which existed prior to its creation. For many reasons, the latter choice is more practical and more likely to be successful. (No amount of “retrying” without other remedial steps is likely to have an effect if, for instance, insufficient storage exists for the instance in the Oracle directories.)

It was reasoned that the best tool for reversing the external effects of a partially completed database instance creation is the segment’s own deinstallation script, provided that it is sufficiently robust to cope with unexpected states. Therefore, the general approach to error handling during the installation of the ELIST Database Instance Segment is to continue past the first detected error (after displaying an appropriate message), performing as many subsequent

steps of the installation process as make sense, and *to leave the segment installed*.⁴ Continuing the installation past the point of the first failure is likely to generate additional messages, calling attention to secondary failures that are direct or indirect consequences of the initial failure. At the conclusion of the installation process, a summary message points out that one or more errors have occurred, that the segment has been left installed, that the installer should consult this document, and that explicitly deinstalling the segment is a suitable way to clean up; it also names the log file created during the installation process, which the installer should consult for additional details about the failure(s), if there is to be any hope of better success on a retry.

The error handling strategy that was implemented is as outlined above, with one modification: Some initial checks are performed before anything is changed, and if these checks fail, the segment is left *uninstalled*. This is reasonable, because no recovery is necessary given that no state changes have occurred yet. The error messages that are displayed in this case clearly state what has happened and what is being done.

It might have been reasonable to have the installation script invoke the deinstallation script automatically after a failure is detected, saving the installer from the need to explicitly deinstall the segment. By *not* doing that, however, we give the installer adequate time to investigate the situation before planning a course of action. In some cases, the installer may also be able to determine that it makes more sense to use *ad hoc* methods (such as SQL scripts or other tools to carry out steps that failed, perhaps with some changes) to proceed in a forward direction, rather than to clean up and remove the partially created instance.

Suggestions that may be helpful for recovering from particular errors during installation are presented later in Section 4.2.1.

4.1.2 General Approach to Error Handling During Deinstallation

Deinstallation of the ELIST Database Instance Segment has the side effect of removing the ELIST database instance (*if* it was created during the installation of the segment, rather than prior to that) and of restoring the data file in the ELIST Global Data Segment in which properties of the instance are recorded to the state it had before the ELIST Database Instance Segment was installed. Though it would be a rare event, removal of the database instance can fail at any of several steps for a variety of reasons. One strategy for dealing with such errors is to leave the segment installed if any are detected, so that presumably the installer can “try again.” However, it is reasonable to expect that subsequent attempts to deinstall the segment will not meet with any more success. Thus, this strategy could lead to a situation in which the segment cannot be deinstalled.

Instead, the following approach was taken. Some initial checks are performed, and failure at this stage indeed leaves the segment still installed. Recovery from these failures is easy and reliable, so no problem is posed by this. Following a failure that occurs after these initial checks have succeeded, the deinstallation process continues to perform subsequent steps that make sense, and the segment is left in a deinstalled state. Both the initial failure and any subsequent failures cause appropriate messages to be displayed, with the final summary message clearly indicating whether or not the segment was deinstalled. Note that failure of one or more deinstallation steps may leave vestiges of the database instance in existence, with no means within the segment

⁴ This last point is somewhat unorthodox; it is more common for segment installation scripts to signal a failure to the COE Segment Installer, causing it to clean up by removing all vestiges of the partially installed segment. If this were to be done, obviously there would be no deinstallation script to use in an attempt to reverse the external effects of the partially completed database instance creation.

(which has been deinstalled) to assist in their cleanup. In such cases, an experienced DBA will be required to employ *ad hoc* methods to clean up the instance. In this case, the final summary message names the log file in which additional clues to the nature of the failure can be found, and it directs the installer's attention to the present document.

The deinstallation script is also designed to be robust enough to be used in an attempt to clean up after a failed installation (see the previous section); this is also true of the ORACLE DataBase Instance (ORADBI) segment, on which it heavily relies. Although it will usually produce a variety of error messages in such a case, it is likely to succeed in its mission, and no further action need be taken.

Suggestions that may be helpful for recovering from particular errors during deinstallation are presented later in Section 4.2.2.

4.2 Handling of Specific Error Situations

4.2.1 Handling of Specific Errors During Installation

This section discusses specific failures that can be diagnosed by error messages during installation of the ELIST Database Instance Segment.

Note that the intent is not to display all possible messages here. Those that arise in *normal* processing are adequately covered in the IP and are not repeated here. In addition, other messages with an obvious meaning, and with no question about what the installer is supposed to do in response, are also not shown here. One example of the latter is a Message Window that is displayed if the installer enters the name of an existing database instance when prompted to enter the name of a *new* (nonexistent) database instance. In this case, the installer is given no option but to close the Message Window, after which the Text Entry Window into which the installer is asked to enter the name of a new database instance is redisplayed.

In fact, there are only two places in the installation process where significant failures possibly requiring corrective action can occur. These correspond to the two major services that the ORACLE DataBase Instance (ORADBI) segment provides to the ELIST Database Instance Segment during the latter's installation. These services are involved with the creation of a new database instance; if the installer elects to use an existing database instance for ELIST, these services are not invoked and therefore cannot fail.

The first service is the registration of the ELIST database instance description with ORADBI, using the `ORADBI_register` API. If a failure occurs here, a message of the following form is displayed:

```
The <instance> database instance has not been created
because ORADBI_register failed to register the ELIST DB
Description; it returned an error code of <status>.
<explanation>
Consult the System Administrator's Manual (SAM) for this
segment. Aborting the install.
```

In all the message templates displayed in this manual, `<instance>` is the name chosen for the new database instance, and `<status>` is the return code from the API under discussion.

If the return code is 1, the `<explanation>` reads as follows:

That error code means that ORADBI_register failed to link the ELIST DB Description into its data directory.

If this occurs, it is presumably an internal error that should be reported to DISA (see Section 8). The segment is left uninstalled, and thus the installer may try again, but of course a second try is likely to fail in the same way.

If the return code is 10, the *<explanation>* reads as follows:

That error code means that the ELIST DB Description has already been registered.

Given that the name of a database description being registered with ORADBI_register is supposed to begin with the unique directory name of the registering segment (see the *System Administrator's Manual (SAM) for ORACLE Client Applications (ORAC), ORACLE DataBase Administration (ORADBA), ORACLE DataBase Instance (ORADBI), and ORACLE RDBMS (ORAS)*), no other segment could have registered a database description that has the same name as the one that *this* segment is trying to register. This segment unregisters the database description during its deinstallation. Therefore, if that database description (which is named ELISTdbinst) is found by ORADBI_register to be already registered, then some kind of failure must have occurred when the ELIST Database Instance Segment was last deinstalled. The circumstances under which the segment can be deinstalled without successfully unregistering the ELISTdbinst database description are difficult to fathom. If this event occurs, it should be reported to DISA.

To recover from this event, a system programmer should execute the following command to unregister ELISTdbinst “manually”:

```
/h/COTS/ORADBI/bin/ORADBI_register '' -r ELISTdbinst
```

The system programmer will probably have to su to root to execute this command successfully. Alternatively, the system programmer can assume the root identity and then remove

- the symbolic link /h/COTS/ORADBI/data/dbdesc/ELISTdbinst,
- the symbolic link /h/COTS/ORADBI/data/ELISTdbinst/data, and
- the directory /h/COTS/ORADBI/data/ELISTdbinst.

The first of the two symbolic links normally points to /h/ELISTdbinst/data/dbdesc, but that directory cannot possibly exist if one is trying to install the ELIST Database Instance Segment (which installs into /h/ELISTdbinst). The second symbolic link normally points to /h/ELISTdbinst/data, which, obviously, also cannot possibly exist.

If the return code is neither 1 nor 10, *<explanation>* is omitted. Scenarios leading to this situation are unknown; consequently, it should be reported (with particulars) to DISA.

The second service is the actual creation of the ELIST database instance, using the ORADBI_auto API of ORADBI. This process fails in a relatively innocuous way if the installer does not succeed in providing a password to be assigned to the SYSTEM account of the

new database instance, when prompted to do so. In this situation, the following message is displayed:

```
You did not successfully enter a password to be
assigned to the SYSTEM account of the new database
instance. Please try again.
```

The installer has no choice but to dismiss the Message Window, whereupon the prompt for the SYSTEM password is repeated.

The process of creating the ELIST database instance can also fail for various reasons after the password prompt is satisfied. If this happens, a message in one of two possible forms is displayed, depending on the severity of the failure and the possibilities for recovery.

The first of the two possible forms is as follows:

```
The <instance> database instance has been created.
However, ORADBI_auto failed to change the SYSTEM
password to the value you supplied.
For further information, consult the System
Administrator's Manual (SAM) for this segment.
```

In this case, the actual instance is intact, but for some unknown reason the password of the SYSTEM account in that instance has not been changed from the initial value (which is manager) to the value the installer provided. The segment is left installed, and the database instance is usable. To change the password to the value you intended it to have, see Section 6.

The second of the two possible forms reads as follows:

```
The <instance> database instance has not been created, and
ORADBI_auto returned an error code of <status>.
<explanation>
Although the instance is not functional, the segment
will remain installed.
This allows information on the failure to be gathered
by studying the log file,
/h/COTS/ORADBI/data/db/<instance>.log.
Deinstalling the segment should successfully remove any
vestiges of the instance.
For further information, consult the System
Administrator's Manual (SAM) for this segment.
```

If the return code is 20, the <explanation> reads as follows:

```
That error code means that Oracle failed to create the
instance.
```

If the return code is anything else, the <explanation> is omitted.

The sentence referring to the log file is included in the message only if the installation proceeded far enough to create the log file; if it exists, the log should certainly be consulted in an attempt to learn more about the nature of the failure. If no log file was created, the sentence referring to the log file is omitted.

As the message indicates, it should be possible to recover from (*i.e.*, clean up after) the failed installation by explicitly deinstalling the segment. (It is conceivable that various error messages will be displayed during its deinstallation, however. If so, the following section can provide assistance.) Depending on the reason for the failure, it may be necessary to increase the size of one or more of the /ora01, /ora02, /ora03, and /ora04 partitions before a subsequent attempt to install the segment can succeed. The required minimum sizes of these partitions are documented in the *ELIST IP*.

4.2.2 Handling of Specific Errors During Deinstallation

This section discusses specific failures that can be diagnosed by error messages during deinstallation of the ELIST Database Instance Segment.

The DEINSTALL script for this segment checks to make sure that other ELIST segments that depend on this one have already been deinstalled; if they have not, an error message states that they must be deinstalled first, and the deinstallation is aborted, leaving this segment installed. In this case, the obvious recovery is to deinstall the dependent segments first, then reattempt to deinstall the ELIST Database Instance Segment.

If the dependent segments have already been deinstalled, then the ELIST Database Instance Segment will positively be deinstalled, even if difficulties are encountered in the process. This design goal makes it possible to remove the ELIST Database Instance Segment under all circumstances. Nevertheless, depending on the nature of the failure, the database instance may remain in place, in whole or in part; on the other hand, it could be successfully removed. The error messages that are displayed should be instrumental in separating these two situations and recovering appropriately if vestiges of the instance do still exist.

Of course, the *installation* of this segment may or may not create a new database instance (see Section 3). A byproduct of the installation of this segment is the recording of information in storage belonging to the ELIST Global Data Segment that identifies the ELIST database instance and indicates whether or not the installation of this segment created it. If a previously existing database instance was identified during the installation of this segment, it is left in place during the deinstallation of the segment (because presumably it is shared by other applications), and in fact the deinstallation cannot fail.

Because the deinstallation logic is heavily dependent on the information recorded in the ELIST Global Data Segment, integrity checks are performed on that information. If the information is missing or garbled, or otherwise cannot be retrieved, the following rather wordy error message is displayed:

```
The Instance_Info file in the ELIST Global Data Segment,
which was written when this segment was installed, is
missing, incomplete, or garbled. The name of the database
instance used for ELIST is unavailable, and furthermore it
is unknown whether that instance existed prior to the
installation of this segment or was created by its
installation. The removal of this segment will proceed, but
without doing anything to delete any database instance. If
there really is an instance to be deleted, a DBA will have
to delete it by methods outlined in the System
Administrator's Manual (SAM) for this segment.
```

Nothing more needs to be done if in fact a pre-existing instance was used. However, if that is not the case, the instance can be removed by using either the **Delete Instances** feature of the **ORADBI** application or the **Db Config Assistant** feature of the **ORAS** application, after logging in to the DBA account mentioned in the “Account Preparation” section of the *ELIST IP*. One should shut the instance down first by using the **Server Control** feature of the **DBAdmR** application. See the *System Administrator’s Manual (SAM) for ORACLE Client Applications (ORAC)*, *ORACLE DataBase Administration (ORADBA)*, *ORACLE DataBase Instance (ORADBI)*, and *ORACLE RDBMS (ORAS)* and the *System Administrator’s Manual (SAM) for Database Administrator Runtime (DBAdmR)*.

A similar situation prevails if the ORACLE DataBase Instance (ORADBI) segment is found to be no longer installed. In this case, the error message reads as follows:

```
The ORACLE DataBase Instance (ORADBI) segment, which is
required, has been removed, hence there is no way for this
DEINSTALL script to delete the <instance> instance. The
removal of this segment will proceed, but without doing
anything to delete the <instance> instance. A DBA will have
to delete it by methods outlined in the System
Administrator’s Manual (SAM) for this segment.
```

It would be very unusual for ORADBI not to be present on the system when the ELIST Database Instance Segment is deinstalled, though it is possible. If ORADBI *is* gone, it is quite possible that the ELIST database instance is also gone, though that is not necessarily the case. An attempt should be made to shut down and then remove the instance as described above (except that the **Delete Instances** feature of the **ORADBI** application is obviously unavailable to help in its removal).

If neither of the preceding messages is displayed, the DEINSTALL script first removes the instance by invoking the ORADBI_auto API of ORADBI, then it unregisters the instance by invoking the ORADBI_register API (both with the -r option). Either of those APIs could fail.

If ORADBI_auto -r fails, a message of the following form is displayed:

```
ORADBI_auto returned an error code of <status> during
the removal of the database instance.
<explanation>
Additional information may be found in the log
file, /h/COTS/ORADBI/data/db/<instance>.log.
Despite the error, the deinstallation of this segment
will continue, and any vestiges of the instance will
probably be cleaned up successfully. For further
information, consult the System Administrator’s
Manual (SAM) for this segment.
```

If the return code from ORADBI_auto (*i.e.*, the <status>) is 30, the <explanation> reads as follows:

```
That error code means that ORADBI_auto failed to
remove the instance.
```

If the return code is anything else, the <explanation> is omitted.

If the deinstallation did not proceed far enough to create a log file, the sentence referring to the log file is omitted.

An attempt should be made to determine whether the instance was, in fact, removed. The “Deinstallation Verification” section for this segment in the *ELIST IP* should be consulted; of course, the log file, if it exists, may also contain helpful information. If it appears that the instance still exists, one may attempt to remove it by using the **Delete Instances** feature of the **ORADBI** application or the **Db Config Assistant** feature of the **ORAS** application, as described previously.

If `ORADBI_register -r` fails, a message of the following form is displayed:

```
ORADBI_register failed to unregister the ELIST DB
Description (it returned an error code of <status>).
<explanation>
Despite the error, the deinstallation of this segment
will continue. For further information, consult the
System Administrator's Manual (SAM) for this segment.
```

If the return code from `ORADBI_register` (i.e., the `<status>`) is 1, the `<explanation>` reads as follows:

```
That error code means that ORADBI_register failed
to unlink the ELIST DB Description.
```

If the return code is anything else, the `<explanation>` is omitted.

To recover from this failure, a system programmer should invoke `ORADBI_register -r` “manually” as outlined (in Section 4.2.1) for recovering from a `<status>` of 10 from `ORADBI_register` when installing the segment. If that fails, report the situation to DISA.

5. Creating an ELIST Database Instance Prior to Installing the ELIST Database Instance Segment

This section provides guidance and instructions for creating a database instance for ELIST prior to installing the ELIST Database Instance Segment, with the goal of making that instance available for the use of the segment when it is installed (instead of letting the segment create a new instance during its installation).

Although the properties of the ELIST database instance that is (or can be) created by the installation of the segment are intended to serve a wide variety of uses, inevitably tradeoffs have to be made, and in rare cases the internally created instance may be inadequate in size or some other characteristic. In these cases it is important to have an alternative way to create the instance, with as much freedom as possible in the choice of its characteristics. In return for this freedom, the installer has to accept a loss of convenience.

Another reason for providing the capability of using an already existing instance is that some operational sites may have a policy that restricts the number of database instances to one, with multiple applications all sharing the single database instance. In that case, the instance that will have to be used for ELIST may have existed long before the administrator chose to install the DII COE ELIST mission application. Although it will not be necessary to use the methods outlined below to create an instance, almost surely some modification of the existing instance will nevertheless be necessary. For example, the ELIST database instance is *required* to have a large, nonstandard rollback segment with a particular name. Thus, at the very least, it will be necessary to add such a rollback segment to the existing instance.

In the remainder of this section, we first discuss the addition of the nonstandard rollback segment to an existing instance; then we discuss the more difficult task of creating an instance for ELIST *de novo*, without installing the ELIST Database Instance Segment. This instance will, of course, include the nonstandard rollback segment.

In addition to any standard public rollback segments, which usually have names like RBS0, RBS1, *etc.*, and which are associated with a tablespace that is usually named RBS, the ELIST database instance must have a public rollback segment named BIG_RBS associated with a tablespace named BIGROLL. The instance that is created by the ELIST Database Instance Segment, when that option is chosen, has four standard public rollback segments, named RBS0 through RBS3, associated with a tablespace named RBS. RBS is made up of three datafiles, each 35 MB in size. By contrast, BIGROLL is made up of a single 500 MB datafile. The full set of parameters pertinent to the standard and nonstandard public rollback segments can be obtained by looking at the printout of the `data/dbdesc/create/SIDrun1.sh` file of the ELIST Database Instance Segment, which may be found in Appendix A of two different documents, the *Software Version Description (SVD) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Instance Segment* and the *ELIST IP*. To add the nonstandard public rollback segment to an existing database instance, log in to the DBA account mentioned in the “Account Preparation” section of the *ELIST IP* and use the **SQL Plus** feature of the **ORAS** application to execute an SQL script that contains a `CREATE TABLESPACE` command for BIGROLL, a `CREATE PUBLIC ROLLBACK SEGMENT` command for BIG_RBS, and an `ALTER ROLLBACK SEGMENT ONLINE` command for BIG_RBS. These commands should include parameters (like sizes) that are *at least* as generous as those in the `data/dbdesc/create/SIDrun1.sh` file.

To create an ELIST instance without installing that segment in such a way that it will be detected as an existing instance when the segment is later installed, one *must* use tools provided by the ORACLE DataBase Instance (ORADBI) segment. In increasing order of difficulty, only the following three options are viable:

- Use the **Create Instances** feature of the **ORADBI** application to invoke the ORADBI Instance Creation Wizard; select a canned instance description (*small*, *medium*, or *large*) that includes sufficiently generous parameters; and then add the nonstandard public rollback segment to the instance after it is successfully created, following the instructions above.
- As in the preceding option, but first use the `make_desc` developer's tool that comes bundled with ORADBI to create a new canned instance, tailored in the appropriate ways, and select *that* instance.

This tool is represented by the `exec/make_desc` script of the ORADBI segment. It is unpolished, and the only documentation available for it is the comments contained within the script.

First use the **Db Config Assistant** feature of the **ORAS** application, with the appropriate options selected, to generate and save the SQL scripts for creating an instance with appropriate properties (including the `BIG_RBS` rollback segment). Then use the `make_desc` tool to add the resulting database description to the ORADBI segment as a new canned instance description. Finally, use the **Create Instances** tool of the **ORADBI** application to create the instance, selecting the new canned instance.

- Develop and install a replacement for the ELIST Database Instance Segment, containing a database description similar to the one in that segment but with parameters modified as appropriate (for example, size parameters may be more generous).

Probably the easiest way to accomplish this is to extract the ELIST Database Instance Segment from the distribution CD without installing it, then modify it as appropriate. Nevertheless, this requires considerable work.

How to perform the foregoing activities in detail is beyond the scope of this SAM. In all cases, it is essential that the DBA read and understand the sections on creating canned or custom instances in the *System Administrator's Manual (SAM) for ORACLE Client Applications (ORAC)*, *ORACLE DataBase Administration (ORADBA)*, *ORACLE DataBase Instance (ORADBI)*, and *ORACLE RDBMS (ORAS)*.

6. Changing the Password of the **SYSTEM** Account of the **ELIST** Database Instance

If the password of the **SYSTEM** account of the **ELIST** database instance needs to be changed after the instance is created, you should use the **Set Password** feature of the **DBAdmR** application. See the *System Administrator's Manual (SAM) for Database Administrator Runtime (DBAdmR)* for instructions on using that feature. The account to which you should log in is the **DBA** account mentioned in the “Account Preparation” section of the *ELIST IP*.

A resourceful **DBA** will be able to achieve the same goal using various features of the **ORAS** application on the database server, or the **ORADBA** application on the application client. The features that can be used to change the **SYSTEM** password are **DBA Studio**, **Security Manager**, and **SQL Plus**. The GUI interfaces for these features all have Help systems. If using the latter, the relevant SQL command is as follows:

```
alter user SYSTEM identified by '<new password>'
```

This page intentionally left blank.

7. Notes

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

Acronym	Definition
API	Application Program Interface
CD	Compact Disk
CM	Configuration Management
COE	Common Operating Environment
COTS	Commercial Off-the-Shelf
DB	Database
DBA	Database Administrator
DBAdmR	Database Administrator Runtime (DII COE segment prefix)
DII	Defense Information Infrastructure
DISA	Defense Information Systems Agency
DTED	Digital Terrain Elevation Data
ELIST	Enhanced Logistics Intratheater Support Tool (DII COE segment prefix)
ETEdit	ETPFDD Editor
ETPFDD	Expanded Time-Phased Force Deployment Data
GB	Gigabyte(s)
GUI	Graphical User Interface
IP	Installation Procedures
KB	Kilobyte(s)
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)
ORAC	ORACLE Client Applications (DII COE segment prefix)
ORADBA	ORACLE Database Administration (DII COE segment prefix)
ORADBI	ORACLE DataBase Instance (DII COE segment prefix)
ORAS	ORACLE RDBMS (DII COE segment prefix)
PC	Personal Computer
RDBMS	Relational Database Management System
SAM	System Administrator's Manual
SQL	Structured Query Language
SVD	Software Version Description

This page intentionally left blank.

8. 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

9. 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

This page intentionally left blank.