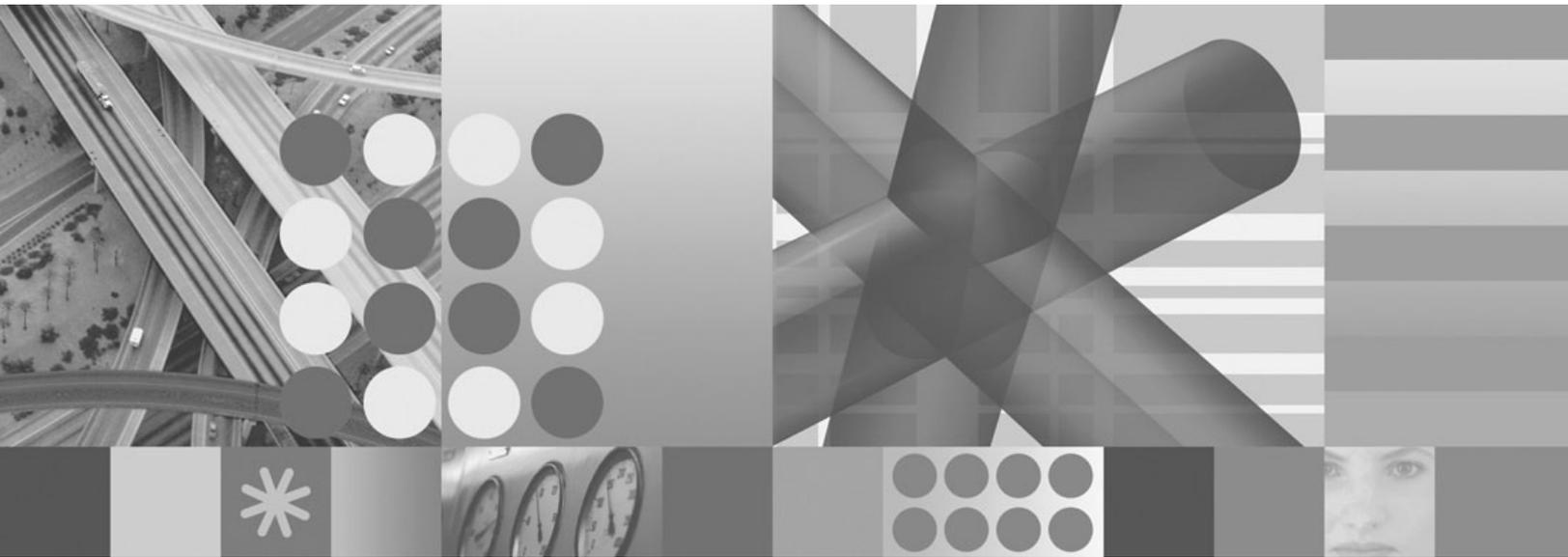




Autonomic Deployment Engine for Administrators



Autonomic Deployment Engine for Administrators

Note

Before using this information and the product it supports, read the information in "Notices" on page 123.

Second Edition (June 2006)

This edition applies to Autonomic Deployment Engine, Version 1.3, and to all subsequent releases and modifications until otherwise indicated.

© Copyright International Business Machines Corporation 2003, 2006. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	v
Who should read this book	v
Related documents	vi
Conventions used in this book	vi

Part 1. Deployment Engine concepts 1

Chapter 1. Overview of Deployment Engine	3
What is Deployment Engine?	3
What can I do with Deployment Engine?	4
What the Deployment Engine technologies do	4
More key terms used by Deployment Engine	5
The components of a Deployment Engine environment	6

Chapter 2. Deployment Engine packages.	11
Software packages	11
Software package types	11
Application packages	12

Chapter 3. The Deployment Engine run-time environment	13
User mode selection for the run-time environment	13
Deployment Engine user modes	13
Multiuser mode	14
Single-user mode	14
Database access	15
Access restrictions in multiuser mode.	15
Access restrictions in single-user mode	15

Part 2. Administration commands 17

Chapter 4. Command summary	19
---	-----------

Chapter 5. Working with commands	21
Command authorization	21
Locating and running the commands	22
Command syntax conventions	22
Case sensitivity in commands	23
Rules for combining command options	23
Specifying an application uniquely	23
Representing strings that include spaces	24
Displaying help for commands	24

Chapter 6. Commands for administering applications	27
de_instmaint	28
de_lsapp	30
de_lsfeat	33
de_lsmaint.	36

de_uninstapp.	38
de_uninstfeat.	40
de_uninstmaint	42

Chapter 7. Commands for administering Deployment Engine . . . 45

de_backupdb	46
de_fs	49
de_lsproc	52
de_restoredb	54
de_setdb	56
de_version.	58

Part 3. Problem determination . . . 59

Chapter 8. Locating the Deployment Engine log files 61

Finding the logger.properties files	61
Finding the log for components	62
Finding the logs for commands.	62

Chapter 9. Message logging 63

Message number	63
Message text	64
Message help.	64
Message log format.	64

Chapter 10. Messages issued by components 67

Common messages	68
Change manager messages	69
Dependency checker messages	70
Installation database messages	71
Operating system touchpoint messages	73

Chapter 11. Messages issued by commands 75

Common messages	76
Main messages	88
de_backupdb messages	95
de_fs messages	97
de_help messages	99
de_instmaint messages	100
de_lsapp messages	101
de_lsfeat messages	102
de_lsmaint messages	103
de_lsproc messages	104
de_restoredb messages	105
de_setdb messages	107
de_uninstapp messages	108
de_uninstfeat messages	109
de_uninstmaint messages	112
de_version messages	113

Chapter 12. Trace logging	115	Open source license notices.	124
Chapter 13. Troubleshooting	117	Apache Software License, Version 1.1	125
Chapter 14. Limitations	119	Apache Software License, Version 2.0	126
<hr/>		W3C Software Notice and License	128
Part 4. Appendixes	121	Trademarks	129
Notices	123	Index	131

Preface

This book describes commands for administering applications that were deployed with the IBM® Autonomic Deployment Engine technology, as well as for administering Deployment Engine itself. It also presents some concepts related to Deployment Engine administration, and provides problem determination information, including messages.

Who should read this book

This book is for administrators who maintain Deployment Engine–enabled applications. A *Deployment Engine–enabled application* is a software application whose installation and maintenance depends on the presence of a Deployment Engine run-time environment on the application's computer. Such software applications are prepackaged for installation and maintenance using Deployment Engine technology and packaging conventions.

This book provides commands for administering a Deployment Engine–enabled application. This book also provides some commands for administering Deployment Engine itself. A user must have special authorities on the application's computer to administer Deployment Engine. System administrators typically have the required authorities to perform any command described in this book, and so, in the broadest sense, this book is addressed to them.

This book is also for general users, who may simply want to maintain a Deployment Engine–enabled application that was installed on their own computer for personal use, as well as for product administrators, who maintain such applications for a wider population such as a user group. If, however, the user is not the system administrator for the computer running the Deployment Engine–enabled application, that user might have to defer to the system administrator to do any local Deployment Engine administration.

For the purposes of this book, general users, product administrators, and system administrators are all considered to be “administrators” and are referred to as such herein—unless the situation requires that a distinction among these users be made.

This book addresses the following tasks, which an administrator can perform on the computer running the Deployment Engine–enabled application:

- Applying maintenance
- Uninstalling applications, features, and maintenance
- Performing file scans on the computer
- Administering the Deployment Engine database (including scheduling Deployment Engine database backups, restoring the database, querying scheduled database backups and file scans, and setting database password and port)
- Querying the Deployment Engine database for inventory information (including installed applications, features, and maintenance)
- Troubleshooting administrator tasks

Note: Some commands and tasks appearing in this book require root, administrator, or other authorities. If so, they are noted as such.

Related documents

In addition to *Autonomic Deployment Engine for Administrators*, Deployment Engine provides the following related documents:

- *IBM Autonomic Deployment Engine: Autonomic Deployment Engine Glossary*
Defines the terminology used in Deployment Engine documentation.

Conventions used in this book

This book uses the following typeface conventions:

Bold

- Commands names that are difficult to distinguish from surrounding text
- Keywords and parameters in text
- Interface controls

Italic

- Words and phrases that are emphasized
- New terms
- Variables and values that you must provide

Monospace

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text, including user prompts
- User input
- Values for arguments or command options

Deployment Engine creates the following environment variables for important directories. In this book, these variable names are sometimes specified with a prepended dollar sign (\$), a UNIX[®] convention, to represent the directories when they appear in the paths of fully qualified file or directory names. For example, `$ACU_COMMON/ACULogger.properties` is used to indicate the location of the `ACULogger.properties` file, regardless of operating system.

ACU_COMMON

The `ACU_COMMON` variable is set to the common directory for Deployment Engine files. This directory is not user-configurable. This directory contains the `ACUApplication.properties` file and the `ACULogger.properties` file. To determine the location of the common directory for Deployment Engine, just query your operating system for the value of the `ACU_COMMON` environment variable.

SI_PATH

The `SI_PATH` variable is set to the directory where Deployment Engine is installed. The default installation directory depends on the command options used when the Deployment Engine run-time environment was installed. To determine the location of the installation directory where Deployment Engine was installed, just query your operating system for the value of the `SI_PATH` environment variable.

This book uses UNIX conventions for specifying environment variables, directory names, or directory paths:

- When using the Microsoft® Windows® command line, replace $\$variable$ with $\%variable\%$ for environment variables or directory names, and replace each forward slash (/) with a backslash (\) in any Windows directory paths that include forward slashes.
- When using the bash shell on Windows operating systems, use the UNIX conventions.

Part 1. Deployment Engine concepts

Chapter 1. Overview of Deployment Engine

This book is for anyone who is maintaining software that was deployed using the technologies of IBM *Autonomic Deployment Engine* (hereafter called *Deployment Engine*).

A note about the Deployment Engine name: The names and content of documentation from earlier releases, as well as some specifications, troubleshooting information, output, files, and the like, refer to Deployment Engine by its previous name, Solution Install for Autonomic Computing, or Solution Install. Occasionally, such references may appear in this book. Also, commands, files, APIs, or other Deployment Engine–related entities might use the characters "si" in their names for the same reason.

This chapter introduces the Deployment Engine approach to software installation and maintenance, or software deployment. *Software deployment* is the process of placing files or installing software into an operating environment, and making that software available for use. As used in this book, software deployment specifically refers to all of the following things, including software removal:

- Installing or removing a software package (where *installing* also involves any initial configuration of the software)
- Applying or removing a feature
- Applying or removing maintenance

A *software package* is a collection of files that includes the application files (referred to as *payload files*) to be deployed plus all the instructions and data that Deployment Engine requires to deploy the application files. Deployment Engine and an installation program—referred to herein as a *software deployment program* because it may not only install but also configure, update, or remove software—both use this software package to deploy all or part of an application in its target environments. An application can be made up of one or more software packages. The software package and its files must have a specific package format and the software package must use designated file-naming conventions.

A *feature* is a set of installable or configuration units that represents some specific functionality of a larger application, and whose deployment is optional. Samples, language packs, or even applications in a suite are considered features.

Maintenance is the general term for any separately deployable software package that represents one or more application updates or migration actions. Fix, incremental update, and full update software packages are specific forms of maintenance.

What is Deployment Engine?

Deployment Engine is a collection of common software packaging, deployment, and configuration technologies from IBM Tivoli®. Tivoli makes these technologies available to IBM and its business partners, whose customers in turn may use it to deploy their applications.

The common technologies provide a *standard way* for application developers to package their software for deployment to a number of popular operating system

and application environments. In the customer environment, Deployment Engine works with the application's software deployment program to orchestrate the software deployment. In addition, Deployment Engine can manage subsequent functionality and maintenance updates on the deployed software, as performed by the administrator.

What can I do with Deployment Engine?

Deployment Engine helps developers and packagers of applications, and the administrators who use and maintain those applications, achieve the following results:

- Install, configure, upgrade, maintain, and remove software, either interactively, silently, or both
- Keep track of installed software
- Manage their prerequisite and corequisite software dependencies
- Manage their environmental dependencies
- Reuse or share the installed software
- Remove applied software upgrades, maintenance, or service packs
- Ensure the tolerance of all dependent software before any shared software is changed or removed
- Create, deploy, and maintain a solution or suite of applications or components
- Handle unsuccessful deployment attempts

To achieve these results, application developers organize their code into logical installable entities, called *installable units*. Installable units are XML representations of the installable parts of an application, whose physical equivalent is the application's payload files. The developers then package the installable units, along with dependency checking and other Deployment Engine infrastructure, to form the final application they deliver to their customers, who install it. At that point, it is usually the administrator's task to provide ongoing maintenance of the installed application and the Deployment Engine infrastructure—or *run-time environment*—that supports the maintenance or removal of the application.

By embedding the Deployment Engine run-time environment in their software, application developers can ensure a more consistent deployment experience for the people who install their software and the administrators who maintain it. Deployment Engine-based deployments require less human interaction and less time to perform. Additionally, the Deployment Engine infrastructure helps reduce the cost and complexity of building, deploying, and maintaining software, solutions, and suites.

What the Deployment Engine technologies do

Deployment Engine by itself does not actually deploy the software. Rather, Deployment Engine works *in concert with* the application's own software deployment program. To this end, Deployment Engine does the following things:

- Defines standardized XML documents that describe installable units of software and their associated requirements, dependencies, and relationships.
- Provides a run-time environment that can consume the standardized XML documents and enable the software deployment.
- Protects the target environment (the environment intended to host the new software) by checking how a new software deployment will affect the software already there. This includes validating the target environment and deployment plan before making any changes to the target environment.

- Uses a standard Web service interface to communicate with target environments and carry out change management operations.
- Provides a database and the appropriate interfaces for logging and tracking deployed software and retaining interdependency and relationship information.

Thus Deployment Engine supplies *the enabling technologies*—the *engine*—for software deployment. These enabling technologies include the standards, facilities, and tools that enable administrators to perform rapid and safe software deployments.

Deployment Engine itself uses open standards like XML, Java™, and Web services so that application developers and administrators can create and deploy installable units which can be managed as elements of their own software application.

More key terms used by Deployment Engine

Before reading the Deployment Engine concepts presented in this chapter, you should understand how Deployment Engine uses the following key terms. The terms as defined here are tailored to Deployment Engine. Some definitions for these terms are further developed later in this book. A complete glossary is provided in a related document, *IBM Autonomic Deployment Engine: Autonomic Deployment Engine Glossary*.

- **Software.** *Software* is any computer programming that provides instructions to the computer hardware to tell it what to do.
- **Application.** *Application* is the term Deployment Engine uses to refer to any software script, component, feature, product, application, application suite, solution, maintenance, or other software deliverable that can be (or already is) deployed in an operating environment.
- **Instance.** A particular occurrence or example of something, such as an *instance* of a deployed application or a feature *instance* in the Deployment Engine installation database.
- **Application package.** Different from a *software package*, as described on page 3, an *application package* is all the software that comprises an application, including any additional application software needed prior to installation or to accomplish installation. It includes all the software packages for the application. It can include Deployment Engine itself, if Deployment Engine is not already part of the operating environment. It can also include other things, such as a Java runtime environment or a software deployment program that an application administrator runs in order to deploy the software packages.
- **Change request.** A *change request* is an object that is passed by the software deployment program to Deployment Engine to request some kind of software change. The change request includes the deployment or configuration operation to be performed as well as the location of the software package. The change request supplies values for variables defined in the software package (such as the value for the installation location). A change request can be associated either with software deployment or with follow-on configuration of the deployed software.

As previously mentioned, Deployment Engine provides an infrastructure for packaging, deploying, and configuring software. Deployment Engine defines these tasks as follows:

- **Packaging.** *Packaging* is the process of reorganizing your application files into a Deployment Engine-compatible software package. This task is usually performed by a *software package developer*.

- **Deploying.** *Deploying* is the process of placing files or installing software into an operating environment and making that software available for use. Deploying, as detailed on page 3, can also involve software configuration and removal. These tasks are usually performed by an *administrator*, a general user, a product administrator, or a system administrator.
- **Configuring.** *Configuring* is the process of setting up or customizing software for a particular use or environment. Unless otherwise stated, *configuring* with Deployment Engine refers to the one-time initial setup of an application or to the subsequent reconfiguration or setup of that application, which can be repeated. Configuration can be designed or automated by the developer who creates the software package, and by the developer who writes the software deployment program. Any real-time configuring is usually performed by the *administrator* who is deploying the application.

The components of a Deployment Engine environment

Note: This is an optional section for system or product administrators who want to understand at a high level the internal componentry of Deployment Engine and its overall environment. (The messages starting on page 67 refer to these Deployment Engine components.) General users who are simply trying to install or maintain their own personal applications are not likely to need the information in this section.

Figure 1 on page 7 shows the key Deployment Engine components (middle of figure) that application developers provide with their Deployment Engine–enabled software. These components are installed on the target computer together with the Deployment Engine–enabled application when that application is installed for the first time.

Also shown are some resources that are associated with Deployment Engine (on the right). One or more of these resources are also located on the target computer where Deployment Engine deploys software. Resources in the target environment that are related to Deployment Engine include managed resources and touchpoints. Application components (on the left)—components that are related to but not provided by Deployment Engine—include the software deployment program, descriptors, and payload files.

The arrows in the figure show how the components developed for the application itself (on the left) and the resources associated with Deployment Engine (on the right) communicate with the actual Deployment Engine components (in the middle). All of the components and resources shown in Figure 1 on page 7 reside on the same computer. They are each described in the text that follows the figure.

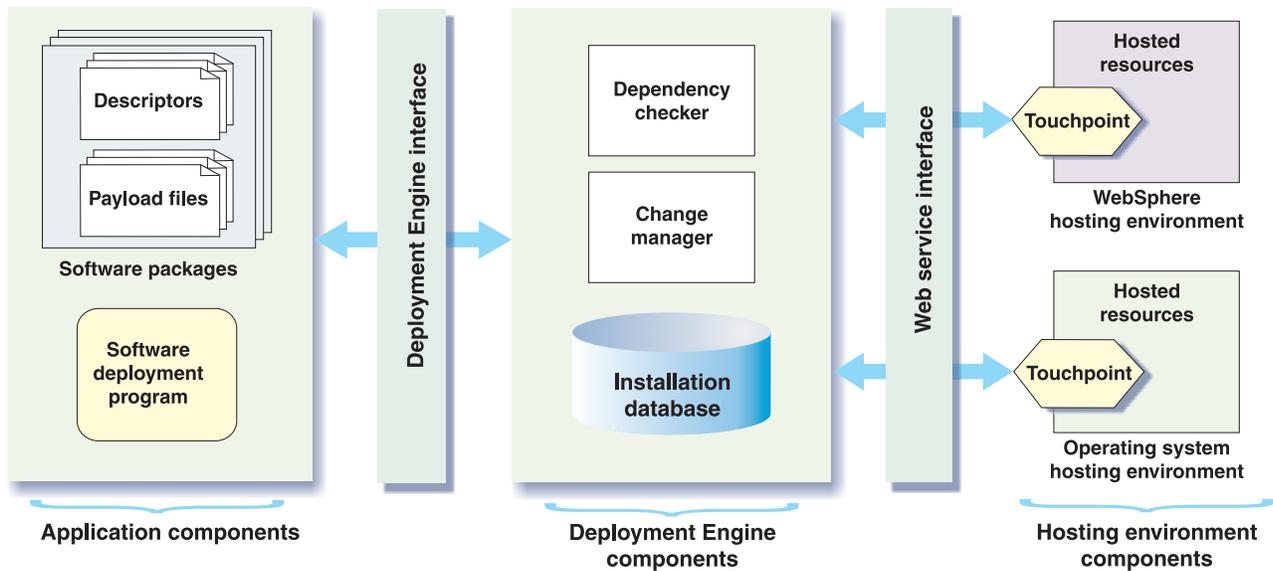


Figure 1. Deployment Engine operating environment on a single computer

A brief description of the key components found in a Deployment Engine operating environment follows. Some components are supplied by Deployment Engine. Other components, noted parenthetically as “application-defined,” are defined or supplied by the application that includes Deployment Engine. Associated resources that are found in the target environments, known as *hosting environments*, are also included in the following list. Each associated resource is noted parenthetically as a “hosting-environment resource.”

Software package (application-defined)

A collection of files that includes the payload files to be deployed plus all the descriptors that Deployment Engine requires to deploy the payload files. Deployment Engine and a software deployment program use this software package to install and configure an application in target environments. An application can be made up of one or more software packages.

Descriptors (application-defined)

XML files that contain the instructions and data that Deployment Engine requires to deploy and configure an application. In Figure 1, “Descriptor” is a general term that represents any of the following XML descriptor files:

- *Installable unit deployment descriptor (IU deployment descriptor)*. An XML descriptor file that defines the content of a software package. The file content includes logical installable software entities, called *installable units* (or *IUs*). A software package contains exactly one IU deployment descriptor.
- *Configuration unit deployment descriptor (CU deployment descriptor)*. An XML descriptor file that defines a single configuration task, though the task might include multiple steps. The file content includes the basic entities of configuration, called *configuration units*, or *CUs*. There can be more than one CU deployment descriptor in a software package.
- *Action descriptor*. An XML descriptor file that defines specific actions for deploying installable units or configuring software in a particular hosting environment. An *action* is simply a task that needs to be performed in a hosting environment.

- *Media descriptor*. An XML descriptor file that identifies the media location of one or more action descriptors or deployable payload files for a software package. A media descriptor is optional. When repackaging software, the media descriptor provides the new media locations for files that have moved.

Payload files (application-defined)

The files in a software package that are deployable; that is, the application files. These are the files that Deployment Engine actually *installs* in the hosting environment. JAR files, ZIP archives, RPM packages, and configuration files are among the valid types of payload files. The actual data content of these payload files is immaterial to Deployment Engine.

Software deployment program (application-defined)

An interactive or silent “installation” program that uses Deployment Engine APIs to first deploy Deployment Engine (if needed) by running a bootstrap program and then implement software change requests in one or more hosting environments. The software deployment program constructs a software change request and passes it to Deployment Engine for processing. This processing then initiates the changes to one or more hosting environments. Developers can code the software deployment program themselves or use tooling to help generate it. Note that, differently from an installation program, a software deployment program not only installs but also configures, updates, or removes software, thereby handling any type of software deployment described in this book.

Deployment Engine interface

Java application programming interfaces (APIs) that the developers of software deployment programs use to generate the change requests that Deployment Engine requires to initiate changes to one or more hosting environments.

Dependency checker

The Deployment Engine component that determines whether or not dependencies are met before installing software in a hosting environment. A *dependency* is a requirement that one installable unit has on another installable unit or managed resource to ensure that they interoperate correctly.

The dependency checker performs *dependency checking* by using data from the following sources to determine whether the dependencies are met:

- The IU deployment descriptor
- The installation database
- The touchpoint for the target hosting environment

The dependency checker also makes sure that the dependencies of other deployed installable units registered in the installation database are not violated. This function is sometimes referred to as *integrity checking*.

Change manager

The Deployment Engine component that reads the information in a deployment descriptor and then coordinates the change request to be implemented across hosting environments.

Installation database

The Deployment Engine component that retains the information about, and the IU deployment descriptor for, each installable unit instance deployed on the local computer. The installation database also contains a *relationship registry* that saves information about the relationships and

interdependencies among deployed installable units. The installation database provided with Deployment Engine is IBM Cloudscape™. Deployment Engine administrators can manage the installation database by using the commands described in Chapter 7, “Commands for administering Deployment Engine,” on page 45.

Web service interface

Represents all Web service communications between the Deployment Engine components and the hosting environments.

Hosting environments

Any environments where Deployment Engine can deploy software (for example, the target operating system or the target J2EE server). Deployment Engine communicates with a hosting environment through its touchpoint.

Hosted resources (hosting-environment resource)

Any resources that are present in the hosting environment and accessible through a touchpoint.

Touchpoints (hosting-environment resource)

Web services that interface with hosting environments; for example, with an *operating system hosting environment* or with a *WebSphere® hosting environment*. A hosting environment has one touchpoint. The touchpoint enables management operations—in this case, operations or processes related to software deployment—to be performed on the hosting environment and on any of its hosted resources.

More about the *operating system touchpoint*: An operating system touchpoint uses *scanners* and *collectors* (collectively called *sensors*) on the target computer to gather information about it. For example, the touchpoint can scan for an application that was installed outside of Deployment Engine, which might satisfy a new application's software requirements. Or the touchpoint can scan for a Deployment Engine–installed application so that Deployment Engine can update it.

The operating system touchpoint can also read deployment actions that originate from an action descriptor in a software package and then use its *effectors* to process the actions in the hosting environment. Actions include creating or removing directories, installing or removing application files, and updating registries, configuration properties, environment variables, paths, and the like.

Chapter 2. Deployment Engine packages

Deployment Engine–based applications include one or more software packages, which can be of various types, in the overall application package. The following sections explain these package-related concepts.

Software packages

A *software package* is a collection of files that includes the application payload files and some descriptors—one deployment descriptor, at least one action descriptor, and often a media descriptor as well. Because the software package represents an application and the instructions for deploying it, the software package is a required component of the overall application package (see page 12). One or more software packages are included in the application package, depending on the complexity of the application (which could, for example, be a solution or suite).

Software package types

Software deployment places application payload files in a hosting environment. The payload files are part of a software package which also includes the descriptors that Deployment Engine requires to deploy the payload files.

A deployable application has at least one, and often more than one, software package. But software packages have different functions. Their function depends on whether the payload files to be deployed are part of the initial version of the application or, alternatively, are some form of application maintenance. Recall that maintenance-related software packages represent one or more application updates in the form of fixes, incremental updates, or full updates.

Therefore, a software package always represents one of the following (either in total or in part):

- The original version of an application
- An full application update
- An incremental application update
- An application fix

As a result, software packages are identified as one of these corresponding types:

Base A software package that contains the original version of an application, called the *base application*. Use a base software package to deploy an application for the first time. A software deployment that installs an application for the first time is referred to as a *fresh installation*.

Full update

A software package that contains a major upgrade to an application, such as a manufacturing refresh. There are two ways to deploy a full update software package: as a fresh installation or as an upgrade.

The software deployment program queries Deployment Engine to determine whether the application is already present in the hosting environment. If the application is already present, the software deployment program can instruct Deployment Engine to deploy the software package as an upgrade. If the application is not present, the software deployment

program instructs Deployment Engine to deploy the software package as a fresh installation. Either deployment yields the same result: a full update of the application to the latest version. A full update software package increases the version number (version, release, modification, or level number) of the application that it updates.

Incremental update

A software package that contains an application upgrade, such as a refresh pack or fix pack. Unlike a full update software package, you cannot deploy an incremental update software package as a fresh installation, because the software package contains software updates only. Instead, you can only deploy it as an upgrade. An incremental update software package is used to upgrade a currently deployed application. An incremental update software package increases the version number (version, release, modification, or level number) of the application that it updates.

Fix A software package that contains critical software changes or corrections (such as an interim fix or test fix) that need to be deployed *sooner* than the next full update or incremental update, which, as a rule, eventually incorporates the same changes provided by the fix. A fix software package is used to deploy only the changed or corrective software and apply it to a currently deployed application.

A fix software package has no impact on the version number of the application that it fixes.

Specific XML elements in the IU deployment descriptor are used to identify these software package types. The **identity** or **fixIdentity** elements for the root IU characterize the type of software package overall. As a rule, each subordinate unit—any installable unit or configuration unit that can be defined within another installable unit—must have identity information for the software package type that matches its root IU. (A full update, however, is the exception; its installable units can have one of two software package types—base or full update.)

In terms of descriptor processing, then, Deployment Engine determines software package types at the installable unit level, in order to deploy the installable units of the software package correctly and efficiently. Ultimately this means that payload files are either installed anew, installed over existing files, or sometimes not installed at all, depending on the software package types associated with their corresponding installable units.

Application packages

An *application package* is all the software packaged as part of the application. It includes all the software packages for the application. It always includes Deployment Engine itself, if Deployment Engine is not, or might not, be part of the operating environment where the application will be deployed. An application package can also include other things, such as a Java runtime environment or a software deployment program that a user runs in order to deploy the software packages.

Chapter 3. The Deployment Engine run-time environment

When an administrator deploys any new Deployment Engine-enabled application, the Deployment Engine run-time environment is automatically installed first in the target installation environment. Its inclusion in the application package is required, unless the appropriate version of the Deployment Engine run-time environment is already known to be present in the installation environment (for example, when deploying maintenance software, the correct version of the run-time environment may already be present from the *initial* software deployment). The application's software deployment program runs the Deployment Engine bootstrap program to check for Deployment Engine in the installation environment and installs it if it is not already there.

User mode selection for the run-time environment

At the time the Deployment Engine run-time environment is installed (that is, at application install time), Deployment Engine examines the user's (or administrator's) authorities and determines which of two user modes to install Deployment Engine in. The particular user mode is determined prior to Deployment Engine installation and cannot be changed following that installation.

A privileged user—a user with special authorities like root or administrator—gets a mode of Deployment Engine (multiuser mode) that provides some extra functionality. Once this Deployment Engine is installed on the computer, it is available to all users of that computer and includes additional capabilities in the areas of scheduling and database access.

A general user—a user with lesser or no special authorities—gets a mode of Deployment Engine (single-user mode) for personal use only, without the privileged-user extras.

The selection of the user mode is transparent to the person who is installing Deployment Engine. But, because the user mode is authority-based, it does affect who can upgrade the installed run-time environment later, what scheduling and database capabilities are available, and, in some cases, what commands the user can perform.

You can determine the user mode that your copy of Deployment Engine was installed in by issuing the **de_version** command, described on page 58. The **de_version** command displays the Deployment Engine user mode as well as the version.

Deployment Engine user modes

When Deployment Engine is transparently installed as part of an application deployment, the authorities of the person performing the installation determine a user mode that is permanently established for the Deployment Engine run-time environment. Selection of the user mode is also transparent to the user or administrator who is installing Deployment Engine as part of the new application. During its initial installation, Deployment Engine determines the user mode based on whether the deploying user is a root or nonroot user.

The sections that follow describe the Deployment Engine user modes, what authorities they require (if any), and the subsequent functionality differences for the run-time environment as a result of those authorities.

Multiuser mode

Multiuser mode is an installation condition where Deployment Engine is installed on a computer and is available to all users of that computer for the purpose of deploying other software. Multiuser mode is established whenever the installer of Deployment Engine has the required authorities for their operating system. Deployment Engine refers to a user with these required authorities as a *root user*. Example root users are:

On Windows operating systems

A root user is a user who has the authority to create a service and write to the Program Files directory. For example, any member of the Administrators group is a root user (provided the group's default permissions were not changed).

On UNIX-based operating systems

A root user is a user who has a user ID of zero (UID=0).

On OS/400® operating systems

A root user is a user who has all of the following authorities: *ALLOBJ, *SECADM, *JOBCTL, and *IOSYSCFG.

A system administrator is almost always a root user.

A multiuser deployment of the Deployment Engine run-time environment provides some additional functionality that is made possible by the operating system authority of the deploying root user. This functionality includes the use of an operating system service that is registered during the initial deployment to enable future scheduling of file system scans and database backups (see Chapter 7, "Commands for administering Deployment Engine," on page 45). This functionality also includes the ability for multiple users to simultaneously access the installation database. Only one multiuser deployment of the Deployment Engine run-time environment can reside on the same computer.

Single-user mode

Single-user mode is an installation condition where a private copy of Deployment Engine is installed on a computer for the purpose of deploying other software. This copy of Deployment Engine in single-user mode is meant for the installer's personal use only (although root users can use it also). Single-user mode is established if the installer of Deployment Engine does not have the required root user authorities for their operating system. Deployment Engine refers to such a user as a *nonroot user*. Example nonroot users are:

On Windows operating systems

A nonroot user is a user who does not have all the root user authorities for their operating system.

On UNIX-based operating systems

A nonroot user is a user who does not have all the root user authorities for their operating system.

On OS/400 operating systems

A nonroot user is a user who does not have all the root user authorities for their operating system, but *does have* the following authorities: *RSTOBJ, *STRTCPSVR, *ENDTCPSVR, and *RSTLIB.

A general user or product administrator is typically a nonroot user.

A single-user deployment of the Deployment Engine run-time environment does not include the additional scheduling and database functionality provided with a multiuser deployment, because the installing nonroot user lacked the operating system authority necessary to provide that functionality. One or more single-user deployments of the Deployment Engine run-time environment can reside on the same computer. However, each user of that computer can deploy only one.

Note: You can have one multiuser and one or more single-user deployments on the same computer, but the software deployment program will not install a Deployment Engine run-time environment in single-user mode if a compatible multiuser deployment already exists.

Database access

The following sections describe the different database access restrictions for Deployment Engine in multiuser mode and in single-user mode.

Access restrictions in multiuser mode

The Deployment Engine installation database can be concurrently accessed. The following database restrictions apply to concurrent access:

- Only one change request is allowed at a time. This restriction exists even if it is different JVMs that are attempting concurrent change requests. If concurrent change requests are attempted, an exception is thrown.
- Any number of dependency checker operations are allowed at any given time. Dependency checker results might be invalidated if the change manager component starts to concurrently process a change request.
- Touchpoints will throw an exception if concurrent action descriptor processing is attempted.

Access restrictions in single-user mode

Because of limitations with the Deployment Engine installation database, there is no concurrent access to the database in single-user mode. Only one user at a time can access Deployment Engine and its related data.

Part 2. Administration commands

Chapter 4. Command summary

Table 1 summarizes the commands that you can use to administer Deployment Engine-enabled applications:

Table 1. Commands for administering applications.

Command	Purpose	Page
de_instmaint	Deploys maintenance to modify or upgrade a previously installed application.	28
de_lsapp	Displays information about an installed application, and optionally saves that information to a file.	30
de_lsfeat	Displays information about the installed features of an application, and optionally saves that information to a file.	33
de_lsmaint	Displays information about the installed maintenance for an application.	36
de_uninstapp	Removes an installed application.	38
de_uninstfeat	Removes an installed feature.	40
de_uninstmaint	Removes an application's previously deployed maintenance from the computer.	42

Table 2 summarizes the commands that you can use to administer Deployment Engine itself:

Table 2. Commands for administering Deployment Engine.

Command	Purpose	Page
de_backupdb	Initiates an immediate backup or schedules a backup of the current Deployment Engine installation database.	46
de_fs	Schedules a file scan to be performed at a specific time or interval, or unchedules a previously scheduled file scan.	49
de_lsproc	Lists all scheduled Deployment Engine backups and file scans.	52
de_restoredb	Restores the Deployment Engine installation database from an existing copy.	54
de_setdb	Resets the password or changes the port number for the Deployment Engine installation database.	56
de_version	Displays the version and user mode of the installed Deployment Engine run-time environment.	58

Code page changes might be required for some languages: For Russian, Hungarian, Polish, and Czech languages, use the following code pages to correctly display message output on a Microsoft Windows NT[®] or Microsoft Windows 2000[®] system:

- Code page 1251 for Russian
- Code page 1252 for Hungarian, Polish, and Czech

From a DOS command window you can change the code page by entering one of the following commands, as appropriate for your language:

```
chcp 1251
```

or

```
chcp 1252
```

Chapter 5. Working with commands

The following sections contain information to be aware of when working with the commands provided by Deployment Engine.

Command authorization

Use of the commands in this chapter depends on the privileges granted to the current user. A root user requires some additional special privileges that a nonroot user does not have. As noted below, even a nonroot user might need some special privileges to issue commands on some operating systems.

Deployment Engine defines a *root user* by operating system, as follows:

On Windows operating systems

A root user is a user who has the authority to create a service and write to the Program Files directory. For example, any member of the Administrators group is a root user (provided the group's default permissions were not changed).

On UNIX-based operating systems

A root user is a user who has a user ID of zero (UID=0).

On OS/400 operating systems

A root user is a user who has all of the following authorities: *ALLOBJ, *SECADM, *JOBCTL, and *IOSYSCFG.

A system administrator is almost always a root user.

Deployment Engine defines a *nonroot user* by operating system, as follows:

On Windows operating systems

A nonroot user is a user who does not have all the root user authorities for their operating system.

On UNIX-based operating systems

A nonroot user is a user who does not have all the root user authorities for their operating system.

On OS/400 operating systems

A nonroot user is a user who does not have all the root user authorities for their operating system, but *does have* the following authorities: *RSTOBJ, *STRTCPSVR, *ENDTCPSVR, and *RSTLIB.

A general user or product administrator is typically a nonroot user.

The "Authorization" section of each command describes the specific *minimum* privileges that the command requires. Thus if the "Authorization" section states that a nonroot user can issue the command, a root user can issue it also.

Some commands—such as the **de_fs** command, the **de_lsproc** command, and the **de_setdb** command—state in their "Description" section that they can only be issued in a Deployment Engine run-time environment that was installed in multiuser mode (modes are described in "Deployment Engine user modes" on page 13). Only root users can issue these commands. Unless otherwise stated, however, a command can be issued in *either* multiuser mode *or* single-user mode.

If you do not know whether your copy of Deployment Engine was installed in multiuser mode or single-user mode, you can find out by issuing the **de_version** command, described on page 58. The **de_version** command displays the Deployment Engine user mode as well as the version.

Locating and running the commands

The Deployment Engine administration command files are installed on the computer in the following location:

DE_installation_dir/bin

where *DE_installation_dir* is the directory where Deployment Engine was installed. To determine the location of *DE_installation_dir*, you can query your operating system for the value of the *SI_PATH* environment variable. For example, on Windows operating systems, the default value for *SI_PATH* is `C:\Program Files\IBM\Common\ASCI`.

On Windows operating systems, specifying *command_name* (along with any desired command options) from the *DE_installation_dir*/bin directory will run the file *command_name.cmd* (for example, specifying `de_backupdb` or `de_backupdb -unschd` will run the command file `de_backupdb.cmd`). On UNIX-based and OS/400 operating systems, it will run the command file *command_name* (with no file extension).

To run a command from another directory, specify the fully qualified command name (along with any desired command options):

DE_installation_dir/bin/*command_name* [-*option_1* *option_2* ... *option_n*]

Command syntax conventions

The command documentation uses the following special characters to define the command syntax:

- [] Identifies optional command entries. Command entries not enclosed in brackets are required.
- ... Indicates that you can specify multiple values for the previous option.
- | Indicates mutually exclusive information. You can use the option to the left of the separator or the option to the right of the separator. You cannot use both options in a single use of the command.
- Bold** Indicates command or command option text.
- Italic* Indicates a variable.
- Prepends an option.

In the “Options” section for each command, command options are listed in the order that they appear in the command syntax.

For other conventions used in this book, see page vi.

Case sensitivity in commands

Command names are case sensitive when the operating system is case sensitive (for example, on UNIX and Linux[®] operating systems). Command names are not case sensitive when the operating system is not case sensitive (for example, on Windows operating systems).

Options are not case sensitive.

Most values are not case sensitive. However, path values are case sensitive on an operating system that is case sensitive. For example, /user/path and /user/Path can refer to different coexisting paths on UNIX or Linux operating systems, so these strings should be treated as case-sensitive values on these operating systems. On Windows operating systems, paths are not case sensitive; so these strings can be treated as such on Windows.

Values (such as discriminant or fix name values) that are stored in the Deployment Engine installation database *are* case sensitive.

The following table shows command syntax elements and their case sensitivity:

Table 3. Case sensitivity of administration command syntax elements, by operating system.

Command syntax element	Case sensitive on UNIX and Linux operating systems?	Case sensitive on Windows operating systems?
Command names	Yes	No
Options	No	No
Values (in general)	No	No
Values (paths and file names)	Yes	No
Values that are compared to strings stored in the Deployment Engine installation database	Yes	Yes

Rules for combining command options

- Options have no order dependency, so that typing *commandname -option1 -option2* is no different than typing *commandname -option2 -option1*.
- Any string beginning with a dash is evaluated as a option. Any string after a space without a dash is considered a value of the last option until the next string starting with a dash is encountered.
- Mutually exclusive options always result in an error. For example, typing *de_lsapp -help -disc* will result in an error.

Specifying an application uniquely

Usually, fully identifying a target application in your administration command requires specifying two options, such as a discriminant value and a UUID value. For example, in Table 4 on page 24, notice that the target applications require a UUID-discriminant pair to specify the application uniquely:

Table 4. Application instances that require both a UUID and discriminant for unique identification.

Target application instance	UUID	Discriminant
Application_instance_1	11111	C:\Program Files\abc
Application_instance_2	11111	C:\Program Files\xyz
Application_instance_3	22222	C:\Program Files\xyz

With only a UUID specification, Deployment Engine cannot distinguish application_instance_1 from application_instance_2, because their UUIDs are the same. However, by also specifying a discriminant in your command, Deployment Engine can locate the correct application instance by its unique combination of UUID and discriminant (the discriminant in this example is an installation directory path). Similarly, with only a discriminant specification, Deployment Engine cannot distinguish application_instance_2 from application_instance_3, because their discriminants are the same. But by additionally specifying a UUID in your command, Deployment Engine can locate the correct application by its unique combination of UUID and discriminant.

Some administration commands, such as the **de_instmaint** and **de_uninstmaint** commands, allow you to use the **-mfile** option to specify a maintenance file that indicates the file name of the fix or incremental update software package to be used by the command. A fix or incremental update software package already includes the UUID of the target application instance, which Deployment Engine can retrieve. Therefore, when using the **-mfile** option, you do not have to specify the UUID, only the discriminant. Deployment Engine will retrieve the UUID from the specified fix or incremental update software package and combine it with the discriminant that you specify to determine the target application instance.

Representing strings that include spaces

Strings that are supposed to be treated as a single value, but contain spaces, should be enclosed in quotation marks (for example, "C:\Program Files\..."). In this case, single (') and double (" ") quotation mark pairs are equivalent.

Displaying help for commands

There are multiple ways to display help. You can use the **de_help** command to display a list of all available administration commands—which includes both the application-related and Deployment Engine-related commands—and their help information:

```
de_help
```

Alternatively, you can display help about a specific command by entering the command name followed by a **-help** option. For example, to display help for the **de_lsfeat** command, enter:

```
de_lsfeat -help
```

The command output will look like the following:

```
de_lsfeat:
```

This command displays information about the installed features of an application, and optionally saves that information to a file.

Command syntax:

```
de_lsfeat -disc <discriminant> [-uuid <UUID>] [-featid
```

```

                                <featureID_1> <featureID_2> ... <featureID_n>] [-ofile <filename>]
                                [-fmt <format>] [-field <fieldname_1> <fieldname_2> ... <fieldname_n>]
                                [-v | -verbose]
de_lsfeat -uuid <UUID> [-disc <discriminant>] [-featid
                                <featureID_1> <featureID_2> ... <featureID_n>] [-ofile <filename>]
                                [-fmt <format>] [-field <fieldname_1> <fieldname_2> ... <fieldname_n>]
                                [-v | -verbose]
de_lsfeat -help | -h | -?
de_lsfeat -usage

```

Command options:

-uuid <UUID>

Specifies the unique identifier for an application. Although the value <UUID> is unique for each application, it is not unique for each application instance. Therefore, you might have to also specify a discriminant for the application instance, in order to uniquely identify it.

-disc <discriminant>

Identifies a particular installed instance of an application. In Deployment Engine, the value <discriminant> is usually, but not always, the path of the directory where the application is installed. But it is possible for a discriminant to have identifiers other than the installation directory. The **-uuid** and **-disc** options specified together identify an installed instance of an application uniquely.

-featid <featureID_1> <featureID_2> ... <featureID_n>

Specifies one or more feature identifiers, delimited by spaces, indicating which features to query.

-ofile <filename>

Specifies the file name of an output file in which to save the query results. If you do not specify the **-ofile** option, the query results are displayed, but they are not saved to an output file.

-fmt <format>

Specifies the format of the query results. This option supports the following formats: xml, text, html, or csv. If no format is specified, the default value is text.

-v | -verbose

Displays in the query results the field names and associated values for all application features and subfeatures. When a feature includes a subfeature, a separate section is displayed for each subfeature. The section includes one or more subfeature fields, with each field consisting of the name of the field and its associated values. If no subfeatures appear in the query results, it means that the application has no subfeatures.

-field <fieldname_1> <fieldname_2> ... <fieldname_n>

Displays the default field names only, along with their associated values. In the option, indicate one or more field names, delimited by spaces. You can specify any of the following field names: featname, featid, and featinstalldate. If the **-field** option is not specified, all fields are displayed.

-help | -h | -?

Displays help for this command.

-usage

Displays usage information for this command.

You can also display help for a specific command by typing the command name followed by the **-h** or **-?** options.

To display only the command syntax for a specific command, type the command followed by the **-usage** option. For example, to display the command syntax for the **de_lsfeat** command, enter the following:

```
de_lsfeat -usage
```

The output will look like the following:

```
de_lsfeat:
```

This command displays information about the installed features of an application, and optionally saves that information to a file.

Command syntax:

```
de_lsfeat      -disc <discriminant> [-uuid <UUID>] [-featid  
               <featureID_1> <featureID_2> ... <featureID_n>] [-ofile <filename>]  
               [-fmt <format>] [-field <fieldname_1> <fieldname_2> ... <fieldname_n>]  
               [-v | -verbose]  
de_lsfeat      -uuid <UUID> [-disc <discriminant>] [-featid  
               <featureID_1> <featureID_2> ... <featureID_n>] [-ofile <filename>]  
               [-fmt <format>] [-field <fieldname_1> <fieldname_2> ... <fieldname_n>]  
               [-v | -verbose]  
de_lsfeat      -help | -h | -?  
de_lsfeat      -usage
```

Chapter 6. Commands for administering applications

These commands are provided to administer an application that was installed or updated using Deployment Engine. Both nonroot and root users can use the administrator commands described in this chapter. (Additional restrictions may be imposed by the computer operating system, its applications, or its system administrator.)

de_instmaint

Deploys maintenance to modify or upgrade a previously installed application.

Syntax

```
de_instmaint -mfile fixfile_1 fixfile_2 ... fixfile_n | updatefile [-disc discriminant]
[-force]
```

```
de_instmaint -help | -h | -?
```

```
de_instmaint -usage
```

Description

Use the **de_instmaint** command to deploy select kinds of software packages in order to update a Deployment Engine-enabled application that was previously installed on the computer.

Maintenance is the general term for any separately deployable software package that represents one or more application modifications, updates, or migration actions. Fix, incremental update, and full update software packages are specific forms of maintenance.

However, this command does not support full update software packages, nor does this command support migration actions. Further, only one incremental update software package can be deployed at a time, and only one application can be updated at one time. Therefore *maintenance, as described for this command, is limited to one incremental update, or to one or more fixes, that are to be deployed to modify or upgrade a single application.*

Deploying maintenance causes application files in the fix or incremental update software package to be installed on the computer as an upgrade to the target application. In addition, during the deployment process, Deployment Engine registers the newly deployed maintenance in the installation database.

Before deploying maintenance, Deployment Engine performs dependency and integrity checking to ensure that the target application can be successfully updated with the fix or incremental update software package.

If any check fails, the maintenance deployment likewise fails. Deployment Engine logs details explaining any failure and displays an error message to the administrator. In the event of a check failure, the administrator could use a **-force** option, under some conditions, to force the maintenance deployment to continue. However, using the **-force** option is not generally recommended, because follow-on failures or subsequent application problems are likely. Administrators should consult with their application support personnel before using the **-force** option.

If all of the checks pass, maintenance deployment continues through completion and a success message is logged and displayed to the administrator.

Options

```
-mfile fixfile_1 fixfile_2 ... fixfile_n | updatefile
```

Specifies the file name of the software package (or packages) to be deployed as maintenance. The value of the **-mfile** option can be the file name of one or more fix software packages, delimited by spaces, or the file

name of a single incremental update software package. Specify a fully qualified file name value, unless you are running the command locally to the software package being deployed.

-disc *discriminant*

Identifies a particular installed instance of an application. In Deployment Engine, the value *discriminant* is usually, but not always, the path of the directory where the application is installed. But it *is* possible for a discriminant to have identifiers other than the installation directory.

-force Specifies that the fix or incremental update software package be installed regardless of an error indicating that a check has failed. To force maintenance processing despite a failed check is risky. For example, if a check fails because a memory requirement is not met, forcing the deployment to go forward will ultimately cause the maintenance deployment to fail anyway, due to insufficient memory for deployment. Even when the force succeeds, the upgraded application might no longer run correctly, or might not run at all. So use the **-force** option with caution. It is a good practice to consult with your application support personnel before using the **-force** option.

-h | -help | -?

Displays help for this command.

-usage Displays usage information for this command.

Authorization

Nonroot user.

See “Command authorization” on page 21 for a description of nonroot users.

Examples

1. The following command installs maintenance file Demo1_Update1.zip:

```
de_instmaint -mfile Demo1_Update1.zip
```

The output looks like the following text:

```
Please wait ...
Installing maintenance package: Demo1_Update1
.....
Request completed successfully.
```

2. The following command installs maintenance file Demo1_Fix1_Interim_FixA.zip, whose unique discriminant is demo:

```
de_instmaint -mfile Demo1_Fix1_Interim_FixA.zip -disc demo
```

The output looks like the following text:

```
Please wait ...
Installing maintenance package: Demo1_Fix1_Interim_FixA
.....
Request completed successfully.
```

Notes

In some circumstances where requirements, relationships, conditions, or other criteria for a safe and satisfactory deployment are violated or not met, you have the option to ignore the failures and force change request processing to go forward anyway. These criteria are put in place to ensure a successful deployment, so, generally speaking, using force processing is not recommended.

de_lsapp

Displays information about an installed application, and optionally saves that information to a file.

Syntax

`de_lsapp`

`de_lsapp [-uuid UUID] [-disc discriminant] [-ofile filename] [-fmt format] [-v | - | -field fieldname_1 fieldname_2 ... fieldname_n]`

`de_lsapp -help | -h | -?`

`de_lsapp -usage`

Description

Displays information about an installed Deployment Engine–enabled application. The information is displayed on-screen, but can additionally be saved to an output file specified in the command.

Options

-uuid *UUID*

Specifies the unique identifier for an application. Although the value *UUID* is unique for each application, it is not unique for each application instance. Therefore, you might have to also specify a discriminant for the application instance, in order to uniquely identify it (see “Specifying an application uniquely” on page 23 for a detailed explanation).

-disc *discriminant*

Identifies a particular installed instance of an application. In Deployment Engine, the value *discriminant* is usually, but not always, the path of the directory where the application is installed. But it *is* possible for a discriminant to have identifiers other than the installation directory. The **-uuid** and **-disc** options specified together identify an installed instance of an application uniquely (see “Specifying an application uniquely” on page 23 for a detailed explanation).

-ofile *filename*

Specifies the file name of an output file in which to save the query results. If you do not specify the **-ofile** option, the query results are displayed, but they are not saved to an output file.

-fmt *format*

Specifies the format of the query results. This option supports the following formats:

- text
- html
- xml
- csv

If no format is specified, the default value is text.

-v | - Displays query results that additionally include sections of information about the application’s features, subfeatures, and maintenance. Each feature, subfeature, or maintenance section includes one or more fields, with each field consisting of the name of the field and its associated

values. If no features, subfeatures, or maintenance appear in the query results, it means that the application does not have them.

-field *fieldname_1 fieldname_2 ... fieldname_n*

Displays the default fields (application name, version, UUID, and discriminant) plus each additional field that you specify with the **-field** option. A displayed field includes the name of the field and its associated values. In the option, indicate one or more field names, delimited by spaces. You can specify any of the following field names:

- prodbuildlevel
- prodinstalldate
- featname
- featid
- featinstalldate
- fixname
- updateversion
- maintbuildlevel
- maintinstalldate
- mainttype

-h | -help | -?

Displays help for this command.

-usage Displays usage information for this command.

Authorization

Nonroot user.

See “Command authorization” on page 21 for a description of nonroot users.

Examples

1. The following command queries application information and displays the default information about all the Deployment Engine–installed applications:

```
de_lsapp
```

The output looks like the following text:

```
Please wait ...
```

```
Sunday, April 30, 2006 9:54:21 PM EDT Deployment Engine 1.3
```

```
Query:  de_lsapp
```

```
-----  
APPLICATION INFORMATION  
-----
```

```
Name:  Demo1  
-----
```

```
Version:  1.0.0.1
```

```
Discriminant:  demo
```

```
UUID:  9EA9516AF0CED5C15C87BC0766D309E5  
-----
```

2. The following command queries application information and displays it in verbose format:

```
de_lsapp -v
```

The output looks like the following text:

de_lsapp

Please wait ...
Sunday, April 30, 2006 9:55:30 PM EDT Deployment Engine 1.3

Query: de_lsapp -v

DEPLOYMENT ENGINE INFORMATION

Version: 1.3
Build Level: 20060428D

APPLICATION INFORMATION

Name: Demo1

Version: 1.0.0.1
Discriminant: demo
UUID: 9EA9516AF0CED5C15C87BC0766D309E5
Build Level: 2005-02-28T12:59:18.0
Install Date: Apr 30, 2006 9:49:16 PM

Features

Name: Demo Feature 1
Feature ID: DemoFeature1
Install Date: Apr 30, 2006 9:49:16 PM

Maintenance

Version: 1.0.0.1
Build Level: 2005-02-28T12:59:18.0
Maintenance Type: FixPack
Install Date: Apr 30, 2006 9:50:36 PM
Uninstallable: Yes

Fix Name: Fix1_Interim_FixA
Build Level: 2005-04-07T10:27:49.0
Maintenance Type: InterimFix
Install Date: Apr 30, 2006 9:52:35 PM
Uninstallable: Yes

de_lsfeat

Displays information about the installed features of an application, and optionally saves that information to a file.

Syntax

```
de_lsfeat -disc discriminant [-uuid UUID] [-featid featureID_1 featureID_2 ... featureID_n] [-ofile filename] [-fmt format] [-field fieldname_1 fieldname_2 ... fieldname_n] [-v | -]
```

```
de_lsfeat -uuid UUID [-disc discriminant] [-featid featureID_1 featureID_2 ... featureID_n] [-ofile filename] [-fmt format] [-field fieldname_1 fieldname_2 ... fieldname_n] [-v | -]
```

```
de_lsfeat -help | -h | -?
```

```
de_lsfeat -usage
```

Description

Displays information about the installed features of a Deployment Engine-enabled application. The information is displayed on-screen, but can additionally be saved to an output file specified in the command.

Options

-disc *discriminant*

Identifies a particular installed instance of an application. In Deployment Engine, the value *discriminant* is usually, but not always, the path of the directory where the application is installed. But it *is* possible for a discriminant to have identifiers other than the installation directory. The **-uuid** and **-disc** options specified together identify an installed instance of an application uniquely (see “Specifying an application uniquely” on page 23 for a detailed explanation).

-uuid *UUID*

Specifies the unique identifier for an application. Although the value *UUID* is unique for each application, it is not unique for each application instance. Therefore, you might have to also specify a discriminant for the application instance, in order to uniquely identify it (see “Specifying an application uniquely” on page 23 for a detailed explanation).

-featid *featureID_1 featureID_2 ... featureID_n*

Specifies one or more feature identifiers, delimited by spaces, indicating which features to query.

-ofile *filename*

Specifies the file name of an output file in which to save the query results. If you do not specify the **-ofile** option, the query results are displayed, but they are not saved to an output file.

-fmt *format*

Specifies the format of the query results. This option supports the following formats:

- text
- html
- xml

de_lsfeat

- csv

If no format is specified, the default value is text.

-field *fieldname_1 fieldname_2 ... fieldname_n*

Displays the specified field names only, along with their associated values. In the option, indicate one or more field names, delimited by spaces. You can specify any of the following field names:

- featname
- featid
- feainstalldate

If the **-field** option is not specified, *all* fields are displayed.

-v | - Displays in the query results the field names and associated values for all application features *and* subfeatures.

When a feature includes a subfeature, a separate section is displayed for each subfeature. The section includes one or more subfeature fields, with each field consisting of the name of the field and its associated values. If no subfeatures appear in the query results, it means that the application has no subfeatures.

-h | -help | -?

Displays help for this command.

-usage Displays usage information for this command.

Authorization

Nonroot user.

See “Command authorization” on page 21 for a description of nonroot users.

Examples

The following command queries and displays information about an installed feature whose unique discriminant is demo:

```
de_lsfeat -disc demo
```

The output looks like the following:

```
Please wait ...
```

```
Sunday, April 30, 2006 9:56:37 PM EDT Deployment Engine 1.3
```

```
Query: de_lsfeat -disc demo
```

```
-----  
APPLICATION INFORMATION  
-----
```

```
Name: Demo1  
-----
```

```
Version: 1.0.0.1
```

```
Discriminant: demo
```

```
UUID: 9EA9516AF0CED5C15C87BC0766D309E5
```

```
Features  
-----
```

```
Name: Demo Feature 1
```

```
Feature ID: DemoFeature1
```

```
Install Date: Apr 30, 2006 9:49:16 PM  
-----
```

Note: Subfeatures are only shown if they exist. If no subfeatures exist for a feature, the Subfeatures line is omitted from the lines that describe the feature.

de_ismaint

Displays information about the installed maintenance for an application.

Syntax

```
de_ismaint -disc discriminant [-uuid UUID] [-fixname fixname_1 fixname_2 ...  
fixname_n | -updateversion version] [-ofile filename] [-fmt format] [-field fieldname_1  
fieldname_2 ... fieldname_n]
```

```
de_ismaint -uuid UUID [-disc discriminant] [-fixname fixname_1 fixname_2 ...  
fixname_n | -updateversion version] [-ofile filename] [-fmt format] [-field fieldname_1  
fieldname_2 ... fieldname_n]
```

```
de_ismaint -help | -h | -?
```

```
de_ismaint -usage
```

Description

Displays information about the installed maintenance for a Deployment Engine-enabled application.

Options

-disc *discriminant*

Identifies a particular installed instance of an application. In Deployment Engine, the value *discriminant* is usually, but not always, the path of the directory where the application is installed. But it *is* possible for a discriminant to have identifiers other than the installation directory. The **-uuid** and **-disc** options specified together identify an installed instance of an application uniquely (see “Specifying an application uniquely” on page 23 for a detailed explanation).

-uuid *UUID*

Specifies the unique identifier for an application. Although the value *UUID* is unique for each application, it is not unique for each application instance. Therefore, you might have to also specify a discriminant for the application instance, in order to uniquely identify it (see “Specifying an application uniquely” on page 23 for a detailed explanation).

-fixname *fixname_1* *fixname_2* ... *fixname_n*

Specifies one or more fix names, delimited by spaces, to query.

-updateversion *version*

Specifies the version of the installed incremental update to query. The original fix incremental update software package need not be present on the computer.

-ofile *filename*

Specifies the file name of an output file in which to save the query results. If you do not specify the **-ofile** option, the query results are displayed, but they are not saved to an output file.

-fmt *format*

Specifies the format of the query results. This option supports the following formats:

- text
- html

- xml
- csv

If no format is specified, the default value is text.

-field *fieldname_1 fieldname_2 ... fieldname_n*

Displays the specified field names only, along with their associated values. In the option, indicate one or more field names, delimited by spaces. You can specify any of the following field names:

- fixname
- updateversion
- maintbuildlevel
- maintinstalldate
- mainttype

-h | -help | -?

Displays help for this command.

-usage Displays usage information for this command.

Authorization

Nonroot user.

See “Command authorization” on page 21 for a description of nonroot users.

Examples

The following command queries and displays information about an installed fix software package whose unique discriminant is demo:

```
de_lsmaint -disc demo
```

The output looks like the following:

```
Please wait ...
Sunday, April 30, 2006 9:58:39 PM EDT Deployment Engine 1.3
```

```
Query:  de_lsmaint -disc demo
```

```
-----
APPLICATION INFORMATION
-----
```

```
Name:   Demo1
-----
```

```
Version:  1.0.0.1
Discriminant:  demo
UUID:    9EA9516AF0CED5C15C87BC0766D309E5
```

```
Maintenance
-----
```

```
Version:  1.0.0.1
Build Level:  2005-02-28T12:59:18.0
Maintenance Type:  FixPack
Install Date:  Apr 30, 2006 9:50:36 PM
Uninstallable:  Yes
-----
```

```
Fix Name:  Fix1_Interim_FixA
Build Level:  2005-04-07T10:27:49.0
Maintenance Type:  InterimFix
Install Date:  Apr 30, 2006 9:52:35 PM
Uninstallable:  Yes
-----
```

de_uninstapp

Removes a previously installed application from the computer.

Syntax

```
de_uninstapp -uuid UUID [-disc discriminant] [-force]
```

```
de_uninstapp -disc discriminant [-uuid UUID] [-force]
```

```
de_uninstapp -help | -h | -?
```

```
de_uninstapp -usage
```

Description

Removes a previously installed Deployment Engine–enabled application from the computer.

Options

-uuid *UUID*

Specifies the unique identifier for an application. Although the value *UUID* is unique for each application, it is not unique for each application instance. Therefore, you might have to also specify a discriminant for the application instance, in order to uniquely identify it (see “Specifying an application uniquely” on page 23 for a detailed explanation).

-disc *discriminant*

Identifies a particular installed instance of an application. In Deployment Engine, the value *discriminant* is often the path of the directory where the application is located, but it can have other values. The **-uuid** and **-disc** options together specify an installed instance of an application uniquely (see “Specifying an application uniquely” on page 23 for a detailed explanation). When uninstalling an application, specify both a discriminant *and* a UUID, unless one or the other is unique.

-force Specifies that the application be removed regardless of an error indicating that a check has failed. To force application removal despite a failed check is risky. For example, by forcing the application removal to go forward, any software that depended on the removed software may no longer function properly. So use the **-force** option with caution. It is a good practice to consult with your application support personnel before using the **-force** option. If this option is not specified, and there *are* dependent applications, the command returns an error message listing the UUIDs associated with the dependent applications that need to be removed.

-h | -help | -?

Displays help for this command.

-usage Displays usage information for this command.

Authorization

Nonroot user.

See “Command authorization” on page 21 for a description of nonroot users.

Examples

The following command uninstalls the application instance whose unique discriminant is demo:

```
de_uninstapp -disc demo
```

The output looks like the following text:

```
Please wait ...  
The application was successfully uninstalled.
```

Notes

Removing an application results in the removal of all files associated with the application and the removal of the application from the Deployment Engine installation database.

de_uninstfeat

Removes an installed feature from the computer. Any subfeatures associated with the feature being removed are also removed.

Syntax

```
de_uninstfeat -featid featureID_1 featureID_2 ... featureID_n -uuid UUID [-disc  
discriminant] [-auto]
```

```
de_uninstfeat -featid featureID_1 featureID_2 ... featureID_n -disc discriminant [-uuid  
UUID] [-auto]
```

```
de_uninstfeat -help | -h | -?
```

```
de_uninstfeat -usage
```

Description

Removes an installed feature from the computer. Any subfeatures associated with the feature being removed are also removed.

Options

-featid *featureID_1 featureID_2 ... featureID_n*

Specifies one or more feature identifiers, delimited by spaces, indicating which features to remove.

-uuid *UUID*

Specifies the unique identifier for an application. Although the value *UUID* is unique for each application, it is not unique for each application instance. Therefore, you might have to also specify a discriminant for the application instance, in order to uniquely identify it (see “Specifying an application uniquely” on page 23 for a detailed explanation).

-disc *discriminant*

Identifies a particular installed instance of an application. In Deployment Engine, the value *discriminant* is usually, but not always, the path of the directory where the application is installed. But it *is* possible for a discriminant to have identifiers other than the installation directory. The **-uuid** and **-disc** options specified together identify an installed instance of an application uniquely (see “Specifying an application uniquely” on page 23 for a detailed explanation).

-auto

Informs Deployment Engine that the features specified in the command, plus any additional maintenance that is dependent on them, be removed without Deployment Engine prompting you to do so. If this option is not specified, and the command does not indicate all of the interdependent maintenance to be removed, the command returns an error message listing the additional maintenance that needs to be removed.

-h | -help | -?

Displays help for this command.

-usage

Displays usage information for this command.

Authorization

Nonroot user.

See “Command authorization” on page 21 for a description of nonroot users.

Examples

The following command uninstalls DemoFeature1 of the application instance whose unique discriminant is demo:

```
de_uninstfeat -disc demo -featid DemoFeature1
```

The output looks like the following text:

```
Please wait ...
```

```
The features were successfully uninstalled.
```

Notes

Removing a feature results in the removal of all files associated with the feature and the removal of the feature record from the Deployment Engine installation database.

de_uninstmaint

Removes an application's previously deployed maintenance from the computer.

Syntax

```
de_uninstmaint -mfile fixfile_1 fixfile_2 ... fixfile_n | updatefile [-disc discriminant]  
[-auto] [-force]
```

```
de_uninstmaint -fixname fixname_1 fixname_2 ... fixname_n | -updateversion  
version -uuid UUID [-disc discriminant] [-auto] [-force]
```

```
de_uninstmaint -fixname fixname_1 fixname_2 ... fixname_n | -updateversion  
version -disc discriminant [-uuid UUID] [-auto] [-force]
```

```
de_uninstmaint -help | -h | -?
```

```
de_uninstmaint -usage
```

Description

Use the **de_uninstmaint** command to remove a Deployment Engine–enabled application's previously deployed maintenance from the computer.

Maintenance is the general term for any separately deployable software package that represents one or more application modifications, updates, or migration actions. Fix, incremental update, and full update software packages are specific forms of maintenance.

However, this command does not support full update software packages, nor does this command support migration actions. Further, only one incremental update can be removed at a time, and only one application can be targeted at one time. Therefore *maintenance, as described for this command, is limited to one incremental update, or to one or more fixes, that are to be removed from a single application.*

Removing maintenance causes application files from previously installed fix or incremental update software packages to be removed from the computer. In addition, during the removal process, Deployment Engine deletes the references to the removed maintenance from the installation database.

Note: Not all installed fixes or incremental updates can be removed. Use the **de_ismaint** command to determine if the maintenance is uninstalleable (see the examples listing maintenance output on page 31) before issuing the **de_uninstmaint** command. Also be aware that even maintenance designated “uninstalleable” could have dependent maintenance that is *not* uninstalleable. In this case, the targeted maintenance cannot be removed and the **de_uninstmaint** command will fail.

Before removing maintenance, Deployment Engine performs dependency and integrity checking. If any check fails, the maintenance removal likewise fails. Deployment Engine logs details explaining any failure and displays an error message to the administrator. In the event of a check failure, the administrator could use a **-force** option to force the maintenance removal to continue. However, using the **-force** option is not generally recommended, because follow-on failures or subsequent application problems are likely. Administrators should consult with their application support personnel before using the **-force** option.

If the checks pass, processing continues through completion and a success message is both logged and displayed to the administrator.

The administrator can use the **-auto** option to automatically remove other maintenance that might be dependent on the fixes or incremental update targeted for removal.

Options

-mfile *fixfile_1 fixfile_2 ... fixfile_n* | *updatefile*

Specifies the file name of the software package (or packages) to be removed. The value of the **-mfile** option can be the file name of one or more fix software packages, delimited by spaces, or the file name of a single incremental update software package. Specify a fully qualified file name value, unless you are running the command locally to the fix or incremental update software package. Use this option when the original fix packages or incremental update software package is still present on the computer.

Note: Unlike with the **-updateversion** and **-fixname** options, when specifying the **-mfile** option, you do not need to specify a UUID to indicate the target application instance uniquely. Deployment Engine can retrieve the UUID from the fix or incremental update software package that is already on the computer (see “Specifying an application uniquely” on page 23 for a detailed explanation).

-disc *discriminant*

Identifies a particular installed instance of an application. In Deployment Engine, the value *discriminant* is usually, but not always, the path of the directory where the application is installed. But it *is* possible for a discriminant to have identifiers other than the installation directory. The **-uuid** and **-disc** options specified together identify an installed instance of an application uniquely (see “Specifying an application uniquely” on page 23 for a detailed explanation).

-auto Informs Deployment Engine that the fixes or incremental update specified in the command, plus any additional maintenance that is dependent on them, be removed without Deployment Engine prompting you to do so. If this option is not specified, and the command does not indicate all of the interdependent maintenance to be removed, the command returns an error message listing the additional maintenance that needs to be removed.

-force Specifies that the fix or incremental update be removed regardless of an error indicating that a check has failed. To force maintenance processing despite a failed check is risky. For example, by forcing the maintenance removal to go forward, currently deployed software could be returned to a previous (backlevel) version that its dependent software cannot use. Following these changes, any software that depended on the now-backleveled software may no longer function properly. So use the **-force** option with caution. It is a good practice to consult with your application support personnel before using the **-force** option.

-fixname *fixname_1 fixname_2 ... fixname_n*

Specifies the name of one or more installed fixes to be removed. Delimit multiple fix names with spaces. The original fix software packages need not be present on the computer.

de_uninstmaint

-updateversion *version*

Specifies the version of the installed incremental update to be removed. The original fix incremental update software package need not be present on the computer.

-uuid *UUID*

Specifies the unique identifier for an application. Although the value *UUID* is unique for each application, it is not unique for each application instance. Therefore, you might have to also specify a discriminant for the application instance, in order to uniquely identify it (see “Specifying an application uniquely” on page 23 for a detailed explanation).

-h | -help | -?

Displays help for this command.

-usage Displays usage information for this command.

Authorization

Nonroot user.

See “Command authorization” on page 21 for a description of nonroot users.

Examples

The following command uninstalls Fix1_Interim_FixA of the application instance whose unique discriminant is demo:

```
de_uninstmaint -fixname Fix1_Interim_FixA -disc demo
```

The output looks like the following text:

```
Please wait ...
Uninstalling maintenance package: Demo1_Fix1_Interim_FixA
.....
Request completed successfully.
```

Chapter 7. Commands for administering Deployment Engine

These commands are provided to administer Deployment Engine itself. Many of the commands are associated with the Deployment Engine installation database. You must be a root user to use most of the administrator commands in this chapter. The “Authorization” section of each command describes the specific privileges that the command requires.

de_backupdb

Initiates an immediate backup of the current Deployment Engine installation database or schedules a backup to be performed at a specific time and frequency.

Syntax

de_backupdb

de_backupdb -bfile *backupfile*

de_backupdb -time *date_time* [**-freq** *frequency*] [**-bfile** *backupfile*]

de_backupdb -unschd

de_backupdb -help | **-h** | **-?**

de_backupdb -usage

Description

Using this command you can initiate an immediate backup of the Deployment Engine installation database or schedule a backup for a specific time and frequency. To initiate an immediate backup of the installation database, simply enter `de_backupdb` with not options.

Options

-bfile *backupfile*

Specifies the name of the backup file. If you do not specify your backup file name as a fully qualified file name, the backup file will be located in the directory from which you issue the **de_backupdb** command. If you do not specify this option at all, the default location will be *DE_installation_dir*/backupdbs, and the default file name will be a timestamp in the form *YYYYMMDDhhmm*.

Note: To determine the location of *DE_installation_dir*, the directory where Deployment Engine was installed, you can query your operating system for the value of the `SI_PATH` environment variable. On Windows operating systems, the default value for `SI_PATH` is `C:\Program Files\IBM\Common\ASCII`.

For scheduled installation database backups, the default name of the backup file—when the **-bfile** and **-time** options are specified—will be the specified file name appended with the timestamp.

-time *date_time*

(Option available in multiuser mode only.) Specifies the date and time that the backup should start. The specified data and time must be later than the time when you issue the **de_backupdb** command. If a frequency other than once is used (see the **-freq** option), the **-time** option indicates the date and time of the first database backup (if once is used, the **-time** option indicates the date and time of the one-time-only database backup).

Date_time values should be specified in the format *yyyy-mm-ddTth:mm:ss*, where *yyyy* is the year, *mm* is the month (from 01 to 12), *dd* is the day of the month (from 01 to 31), **T** is a constant that separates the date and time,

hh is the hour (from 00 to 23), *mm* is the minute (from 00 to 59), and *ss* is the second (from 00 to 59). When specifying a time for the current date, you only need to specify *hh:mm:ss*.

For example, specify the time January 31, 2006 at 12:00:00 p.m. as follows:
2006-01-31T12:00:00

Specify today at exactly 11 p.m. as follows:
23:00:00

-freq *frequency*

(Option available in multiuser mode only.) Schedules the interval in which the database backup should occur. The following values are valid frequencies:

- once
- daily
- weekly
- monthly

If you do not specify a frequency, the value *once* is used.

-unschd

(Option available in multiuser mode only.) Clears the schedule of all its database backup entries. The database backups associated with the cleared entries will not be performed. A new **de_backupdb** command must be issued to initiate or schedule any new backups.

-h | -help | -?

Displays help for this command.

-usage Displays usage information for this command.

Authorization

For a Deployment Engine run-time environment that was installed in multiuser mode, only a root user can run this command.

For a Deployment Engine run-time environment that was installed in single-user mode, a nonroot user can run this command.

See “Command authorization” on page 21 for a description of root and nonroot users.

If you do not know whether your copy of Deployment Engine was installed in multiuser mode or single-user mode, you can find out by issuing the **de_version** command, described on page 58. The **de_version** command displays the Deployment Engine user mode as well as the version.

Examples

1. The following command backs up the installation database weekly at 4 a.m., starting on May 9, 2006:
de_backupdb -time 2006-05-09T04:00:00 -freq weekly

de_backupdb

The output looks like the following text:

```
Scheduling backup, please wait ...
ACUDB0004I A database backup command was issued, but directory
C:\Program Files\IBM\Common\acsi\backupDBs, the default directory
where database backup information is stored, does not exist.
Deployment Engine will create the directory.
```

```
-----
Process:                Deployment Engine Database Backup
-----
Start date/time:       May 9, 2006 4:00:00 AM
Frequency:             WEEKLY
Backup File:           C:\Program Files\IBM\Common\acsi\backupDBs\200605090400
```

Request completed successfully.

2. The following command backs up the installation database to backup file backup.db in the top-level C directory. The backup occurs once, which is the default frequency:

```
de_backupdb -bfile c:\backup.db
```

The output looks like the following text:

```
Backing up database to C:\backup.db; please wait ...
Request completed successfully.
```

de_fs

Schedules a file scan to be performed at a specific time or interval, or un schedules a previously scheduled file scan.

Note: This command can only be issued in a Deployment Engine run-time environment that was installed in multiuser mode. (See “Multiuser mode” on page 14 for a description of multiuser mode.) If you do not know whether your copy of Deployment Engine was installed in multiuser mode or single-user mode, you can find out by issuing the **de_version** command, described on page 58. The **de_version** command displays the Deployment Engine user mode as well as the version.

Syntax

```
de_fs -time date_time [-freq frequency]
```

```
de_fs -unschd
```

```
de_fs -help | -h | -?
```

```
de_fs -usage
```

Description

Use the **de_fs** command to run a file scan on a specified schedule. You can schedule the file scan for a specific time, either to run once or to run periodically. Periodic runs can be at daily, weekly, or monthly intervals. The file scan looks for installed applications—especially applications that were not installed by Deployment Engine—and registers them in a database for future reference. For example, an application that is known and registered could be used to satisfy a prerequisite software requirement for another application about to be installed.

An operating system touchpoint (described on page 9) uses scanners and collectors on the target computer to gather information about the target computer. In particular, the scanners discover hardware, operating system, networking, software, or other information related to the target computer. The function of one particular scanner is to perform file scans that look for software signatures.

A *file scan* is a background process whereby a specific operating system touchpoint scanner checks all the files on the target computer and registers any applications it finds in a database table. The database belongs to the touchpoint (that is, it is not the installation database that belongs to Deployment Engine). Software signatures provided by the touchpoint determine what applications this scanner can look for.

Before the file scan is processed, the touchpoint clears its existing database table of applications previously found on the computer by the scanner. After the file scan runs, the touchpoint repopulates the database table with the new list of applications found by the scanner. (Applications and information obtained by other touchpoint scanners are not cleared from the database table.)

When Deployment Engine is first installed, a weekly file scan is scheduled on the target computer that starts one week from the installation date at 2:00 a.m. A file scan can take a long time to run if the computer has many files. So rather than running the file scan during software deployment, the scan is scheduled to run later, as a background process. It is important that a file scan runs periodically, to

both detect potential prerequisite applications that might be needed for future software deployments and to detect installations that occurred outside of Deployment Engine.

When and how often the file scan should run depends mainly on the frequency with which software is installed on (or removed from) the computer. Deployment Engine makes the **de_fs** command available so an administrator can customize the schedule as needed. As administrator, this gives you the option to supersede the original weekly file scan and schedule your own instead.

For example, the default 2:00 a.m. time might be inconvenient, or you might want to perform file scanning more frequently to keep your list of known installed software current. You might prefer to schedule a file scan just before a planned software installation, in order to have the latest application information available during deployment. Or you might want to schedule a file scan just after software removal, to have application information that reflects your latest changes.

Be aware that if you unschedule a file scan without scheduling a new one, you effectively turn off file scanning for this particular scanner. Should that happen, Deployment Engine might not have a complete record of all the applications currently installed by Deployment Engine or by other means on the local computer. Thus any future deployment could result in multiple copies of installed applications, or result in a failed deployment because Deployment Engine was unaware of some requisite software that was actually present on the target computer.

Options

-time *date_time*

Specifies the date and time of the file scan. The specified data and time must be later than the time when you issue the **de_fs** command. If a frequency other than once is used (see the **-freq** option), the **-time** option indicates the date and time of the first file scan (if once is used, the **-time** option indicates the date and time of the one-time-only file scan).

Date_time values should be specified in the format *yyyy-mm-ddT***hh:mm:ss**, where *yyyy* is the year, *mm* is the month (from 01 to 12), *dd* is the day of the month (from 01 to 31), **T** is a constant that separates the date and time, *hh* is the hour (from 00 to 23), *mm* is the minute (from 00 to 59), and *ss* is the second (from 00 to 59). When specifying a time for the current date, you only need to specify *hh:mm:ss*.

For example, specify the time January 31, 2006 at 12:00:00 p.m. as follows:
2006-01-31T12:00:00

Specify today at exactly 11 p.m. as follows:
23:00:00

-freq *frequency*

Schedules the interval in which the file scan should occur. The following values are valid frequencies:

- once
- daily
- weekly
- monthly

If you do not specify a frequency, the value once is used.

-unschd

Clears the schedule of all its file scan entries. The file scans associated with the cleared entries will not be performed. A new **de_fs** command must be issued to schedule any new file scans.

-h | -help | -?

Displays help for this command.

-usage Displays usage information for this command.

Authorization

Root user.

See “Command authorization” on page 21 for a description of root users.

As noted previously, this command can only be issued in a Deployment Engine run-time environment that was installed in multiuser mode. (See “Multiuser mode” on page 14 for a description of multiuser mode.) If you do not know whether your copy of Deployment Engine was installed in multiuser mode or single-user mode, you can find out by issuing the **de_version** command, described on page 58. The **de_version** command displays the Deployment Engine user mode as well as the version.

Examples

The following command performs a file scan daily at 3 a.m., starting on May 8, 2006:

```
de_fs -time 2006-05-08T03:00:00 -freq daily
```

The output looks like the following text:

Scheduling file scan, please wait ...

```
-----
Process:                               Deployment Engine File Scan
-----
Start date/time:                        May 8, 2006 3:00:00 AM
Frequency:                              DAILY
```

Request completed successfully.

de_lsproc

Lists all scheduled installation database backups and file scans.

Note: This command can only be issued in a Deployment Engine run-time environment that was installed in multiuser mode. (See “Multiuser mode” on page 14 for a description of multiuser mode.) If you do not know whether your copy of Deployment Engine was installed in multiuser mode or single-user mode, you can find out by issuing the **de_version** command, described on page 58. The **de_version** command displays the Deployment Engine user mode as well as the version.

Syntax

```
de_lsproc
```

```
de_lsproc -help | -h | -?
```

```
de_lsproc -usage
```

Description

The **de_version** command displays all scheduled Deployment Engine installation database backups and file scans. No command options are required, simply enter the following:

```
de_lsproc
```

Options

```
-h | -help | -?
```

Displays help for this command.

```
-usage
```

 Displays usage information for this command.

Authorization

Root user.

See “Command authorization” on page 21 for a description of root users.

As noted previously, this command can only be issued in a Deployment Engine run-time environment that was installed in multiuser mode. (See “Multiuser mode” on page 14 for a description of multiuser mode.) If you do not know whether your copy of Deployment Engine was installed in multiuser mode or single-user mode, you can find out by issuing the **de_version** command, described on page 58. The **de_version** command displays the Deployment Engine user mode as well as the version.

Examples

The following command queries and displays all scheduled installation database backups and file scans:

```
de_lsproc
```

The output looks like the following text:

```
-----  
Process:                Deployment Engine Database Backup  
-----  
Start date/time:       May 9, 2006 4:00:00 AM  
Frequency:             WEEKLY  
Backup File:          C:\Program Files\IBM\Common\acsi\backupDBs\200605090400  
  
-----  
Process:                Deployment Engine File Scan  
-----  
Start date/time:       May 8, 2006 3:00:00 AM  
Frequency:             DAILY
```

de_restoredb

Restores the Deployment Engine installation database from an existing copy.

Attention: Initiating this command *deletes the current Deployment Engine installation database* prior to restoring the database from a saved copy.

Syntax

```
de_restoredb [-bfile backupfile]
```

```
de_restoredb -help | -h | -?
```

```
de_restoredb -usage
```

Description

Using this command you can restore a Deployment Engine installation database from an existing copy of that database. Use this command with care because it will first delete the current Deployment Engine installation database before performing the database restoration.

If no options are entered for the **de_restoredb** command, the restore is performed using the most recent backup of the installation database.

Options

-bfile *backupfile*

Indicates the name of the backup file that should be used to restore the installation database. Specify a fully qualified file name to indicate the location of the backup file, unless you are issuing the command from the directory where the backup file is actually located. The default location is *DE_installation_dir/backupdbs*, and the default file name is a timestamp in the form *YYYYMMDDhhmm*.

Note: To determine the location of *DE_installation_dir*, the directory where Deployment Engine was installed, you can query your operating system for the default value of the *SI_PATH* environment variable. On Windows operating systems, the value for *SI_PATH* is *C:\Program Files\IBM\Common\ASCII*.

For scheduled installation database backups, the default name of the backup file—when the **-bfile** and **-time** options are specified—will be the specified file name appended with the timestamp.

-h | -help | -?

Displays help for this command.

-usage Displays usage information for this command.

Authorization

For a Deployment Engine run-time environment that was installed in multiuser mode, only a root user can run this command.

For a Deployment Engine run-time environment that was installed in single-user mode, a nonroot user can run this command.

See “Command authorization” on page 21 for a description of root and nonroot users.

If you do not know whether your copy of Deployment Engine was installed in multiuser mode or single-user mode, you can find out by issuing the **de_version** command, described on page 58. The **de_version** command displays the Deployment Engine user mode as well as the version.

Examples

The following command restores the installation database from backup file backup.db in the top-level C directory:

```
de_restoredb -bfile c:\backup.db
```

The output looks like the following text:

```
Restoring database from C:\backup.db; please wait ...  
Request completed successfully.
```

Notes

If no options are entered for this command, the restore will be performed using the most recent backup in the *DE_installation_dir*/backupdbs directory.

de_setdb

Resets the password or changes the port number for the Deployment Engine installation database.

Note: This command can only be issued in a Deployment Engine run-time environment that was installed in multiuser mode. (See “Multiuser mode” on page 14 for a description of multiuser mode.) If you do not know whether your copy of Deployment Engine was installed in multiuser mode or single-user mode, you can find out by issuing the **de_version** command, described on page 58. The **de_version** command displays the Deployment Engine user mode as well as the version.

Syntax

```
de_setdb -port port_number [-resetpw]
```

```
de_setdb -resetpw [-port port_number]
```

```
de_setdb -help | -h | -?
```

```
de_setdb -usage
```

Description

Use the **de_setdb** command to reset the password, change the port number, or both for the Deployment Engine installation database.

If you are the administrator of a computer that supports Deployment Engine-enabled applications, and you need to reset the installation database password on a regular basis—for example, to comply with your company's password policy—you can use this command to do so. (The new password is determined solely by Deployment Engine, so you cannot specify one. Also, the password is internal to Deployment Engine, so no new password information is returned.)

You can also use the command to change the port number for the installation database. Deployment Engine assigns a port number to its database based on port availability on the computer at the time the Deployment Engine run-time environment is installed. If subsequent port conflicts arise or you need to change to a particular port number, an administrator can use the **de_setdb** command to change the port number by specifying a new one.

Options

-port *port_number*

Indicates the new port for Deployment Engine to use. Specify a number from 1024 to 65534 for the *port_number* value. Port numbers are assigned by a system administrator.

-resetpw

Causes Deployment Engine to reset its installation database password. The new password is determined by Deployment Engine, so you cannot specify one yourself. The password is internal to Deployment Engine; no new password is returned.

-h | -help | -?

Displays help for this command.

-usage Displays usage information for this command.

Authorization

Root user.

See “Command authorization” on page 21 for a description of root users.

As noted previously, this command can only be issued in a Deployment Engine run-time environment that was installed in multiuser mode. (See “Multiuser mode” on page 14 for a description of multiuser mode.) If you do not know whether your copy of Deployment Engine was installed in multiuser mode or single-user mode, you can find out by issuing the **de_version** command, described on page 58. The **de_version** command displays the Deployment Engine user mode as well as the version.

Examples

The following command changes the port number for the Deployment Engine installation database to port 23495:

```
de_setdb -port 23495
```

The output looks like the following text:

```
Changing the database port; please wait ...  
Request completed successfully.
```

de_version

Displays the version and user mode of the installed Deployment Engine run-time environment.

Syntax

```
de_version
```

```
de_version -help | -h | -?
```

```
de_version -usage
```

Description

Displays the version number of Deployment Engine, and whether the Deployment Engine run-time environment was installed in single-user mode or multiuser mode. Some commands for administering Deployment Engine require different authorities, depending on the mode in which Deployment Engine was installed. For details about these modes, see “Deployment Engine user modes” on page 13.

Options

```
-h | -help | -?
```

Displays help for this command.

```
-usage
```

 Displays usage information for this command.

Authorization

Nonroot user.

See “Command authorization” on page 21 for a description of nonroot users.

Examples

The following command queries and displays version and user mode information for Deployment Engine:

```
de_version
```

The output might look like the following:

```
Apr 30, 2006
```

```
Deployment Engine, version 1.3 (build 20060428D) (multiuser mode)
```

Part 3. Problem determination

Chapter 8. Locating the Deployment Engine log files

Deployment Engine logs message and trace information into flat-text log files. These log files can be used for troubleshooting purposes. The locations of the Deployment Engine message and trace logs for either commands or components can be found in the `logger.properties` files.

To locate the Deployment Engine logs:

1. Find the `logger.properties` files, as described on page 61.
2. Find the logs for components, as described on page 62.
3. Find the logs for commands as described on page 62.

Finding the `logger.properties` files

To determine the location of the Deployment Engine message and trace log files, you need to look in the following properties files:

- The `ACULogger.properties` file (for logs related to the Deployment Engine components)
- The `deexlogger.properties` file (for logs related to the Deployment Engine commands)

These properties files can be found in the common directory (referred to as `$ACU_COMMON`) for Deployment Engine files.

If the computer is running Deployment Engine in multiuser mode, the common directory is located as follows:

Windows operating systems

`C:\Program Files\ibm\common\acsi`

UNIX-based operating systems

`/var/ibm/common/acsi`

OS/400 operating systems

`/QOpenSys/QIBM/ProdData/acsi`

If the computer is running Deployment Engine in single-user mode, the common directory is located as follows, where *username* is the user ID of the user on the current operating system:

Windows operating systems

`C:\Documents and Settings\username\acsi_username`

UNIX-based operating systems

`/home/username/.acsi_username`

OS/400 operating systems

`/home/username/.acsi_username`

After locating the `logger.properties` files for components and commands, use the information in the following sections to find the log files for messages and traces.

Finding the log for components

In the sample property-file lines shown below, a **fileName** line specifies the file name for a log file. The subsequent **fileDir** line specifies the path of the directory, *username*, where that log file is located. The name of the subdirectory *username* matches the user ID of the user on the current operating system. In the **fileDir** line, *DE_installation_dir* represents the top-level directory where Deployment Engine was installed.

To find the message log file for the Deployment Engine component messages, look for the following lines in the `$ACU_COMMON/ACULogger.properties` file:

```
acu.message.handler.file.fileName = si_msg.log  
acu.message.handler.file.fileDir = DE_installation_dir/logs
```

Note: The **fileDir** line represents the path to the log file directory, *username*, where *username* is the user ID of the user on the current operating system. Therefore the log file will actually be located in *DE_installation_dir/logs/username*.

Finding the logs for commands

In the sample property-file lines shown below, a **fileName** line specifies the file name for a log file. The subsequent **fileDir** line specifies the path of the directory, *username*, where that log file is located. The name of the subdirectory *username* matches the user ID of the user on the current operating system. In the **fileDir** line, *DE_installation_dir* represents the top-level directory where Deployment Engine was installed.

To find the message log file for the Deployment Engine command messages, look for the following lines in the `$ACU_COMMON/deexlogger.properties` file:

```
com.ibm.ci.log.msg.fileName=deex_msg.log  
com.ibm.ci.log.msg.fileDir=DE_installation_dir/logs
```

Note: The **fileDir** line represents the path to the log file directory, *username*, where *username* is the user ID of the user on the current operating system. Therefore the log file will actually be located in *DE_installation_dir/logs/username*.

To find the trace log file for the Deployment Engine command messages, look for the following lines in the `$ACU_COMMON/deexlogger.properties` file:

```
com.ibm.ci.log.trace.fileName=deex_trace.log  
com.ibm.ci.log.trace.fileDir=DE_installation_dir/logs/username
```

Note: The **fileDir** line represents the path to the log file directory, *username*, where *username* is the user ID of the user on the current operating system. Therefore the log file will actually be located in *DE_installation_dir/logs/username*.

Chapter 9. Message logging

The sections that follow explain the standard form of Deployment Engine messages and the format of the message logs.

Message number

Deployment Engine message numbers have the following format:

AAABnnnnC

where the parts of the message are as follows:

AAA The "product" prefix. The prefix for Deployment Engine is **ACU**.

BB The "component" prefix. The "component" prefix for Deployment Engine commands is **EX**. The component prefixes Deployment Engine subcomponents, or internal components, are as follows:

SI Common (indicates messages common to multiple internal components)

DB Installation database component

CM Change manager component

DC Dependency checker component

OS Operating system touchpoint

nnnn A numeric identifier unique within the combination of product and component prefixes.

C The severity code indicator:

I **Informational:** Informational messages provide users with information or feedback about normal events that have occurred or are occurring, or request information from users in cases where the outcome will not be negative, regardless of the response.

Examples:

- The status request is processing.
- The files were successfully transferred.
- Do you want to save your output in file a or in file b?

Note: Informational messages issued by Deployment Engine are not documented in this chapter, as they are complete in themselves and require no further information or explanation. This also applies to informational messages giving the usage of the Deployment Engine administrator commands.

W **Warning:** Warning messages indicate that potentially undesirable conditions have occurred or could occur, but the program can continue. Warning messages often ask users to make decisions before processing continues.

Examples:

- A requested resource is missing. Processing will continue.

- A file already exists with the same name. Do you want to overwrite this file?

E **Error:** Error messages indicate problems that require intervention or correction before the program can continue.

Examples:

- The specified file could not be found.
- You are out of space on the x drive. The file cannot be saved to this drive.

Message text

Every attempt has been made to represent the message text exactly as it appears in the displayed or written message.

Where the system has included variable information in the message text, this variable information is represented by an italicized label, describing the type of information referred to by the variable. For example, if the message text that appears on your screen is:

The error code is 1.

the message text shown in this chapter would be:

The error code is *error_code*.

In this case the label *error_code* tells you that the information that will be inserted into the message by Deployment Engine is the error code for the problem.

Message help

In addition to the message text itself, help information is provided for most warning and error messages. In this book, message help can include the following help topics:

Explanation

This help topic typically expands on the information provided in the message text, more fully explaining the circumstances in which the message was issued.

User Response

This help topic tells the user what to do next. This might include instructions on how to solve a problem or correct an error. Sometimes, particularly with warning messages, there is nothing that the user needs to do. Sometimes the user is directed to where to find more information, or, if necessary, who to see for further assistance.

Message log format

The message log for components and the message log for commands (whose names and locations are determined using the information starting on page 61) each contain a list of messages for Deployment Engine administrators. These logs include the following fields:

Message entry date

Indicates the year, month, and day that the message entry was generated.

Message entry time

Indicates the time of day that the message entry was generated.

Java class name

Indicates the name of the Java class that generated the message entry.

Method name

Indicates the name of the method that generated the message entry.

Host name

(This field is available only in the message log for components.) Indicates the fully qualified host name of the computer that is running the instance of Deployment Engine that generated the message entry.

Trace logging level

(This field is available only in the message log for commands.) Indicates the Java-defined standard logging level for the message entry. The default message logging level is CONFIG. (See page 115 for a list of the valid trace logging levels, which are also valid for messages.)

Message identifier

Indicates the identifier of the message entry.

Message text

Indicates the text of the message entry.

Chapter 10. Messages issued by components

This chapter contains message help information for all of the error and warning messages issued by internal components of Deployment Engine that are referenced by a unique message reference number. The Deployment Engine components that issue these messages are described on page 6.

The messages are presented in the order listed below. The Deployment Engine common messages are presented first. Then the messages for each Deployment Engine internal component are presented in alphanumeric order, beginning with the change manager subcomponent.

- “Common messages” on page 68
- “Change manager messages” on page 69
- “Installation database messages” on page 71
- “Dependency checker messages” on page 70
- “Operating system touchpoint messages” on page 73

Note: Informational messages issued by Deployment Engine are not documented in this chapter, because they are complete in themselves and require no further information or explanation. This also applies to informational messages that provide usage information about Deployment Engine commands.

Common messages

The messages in this section are common to more than one Deployment Engine component.

ACUSI0000E A processing error occurred that you cannot resolve by yourself. The error code is *error_code*.

Explanation: This error happened within internal componentry that is not generally accessible. Fixing the problem requires outside assistance.

User response: Make a note of the error code, and contact your support representative for help.

ACUSI0002E The Deployment Engine session is currently busy processing another operation. The new operation cannot be processed.

Explanation: Deployment Engine must process database operations singly to prevent the corruption of its database. Because one such operation was already in progress, the new operation was not processed.

User response: Make sure you are not doing multiple operations that affect the database at the same time. Examples of multiple operations include simultaneous software deployments and a software deployment that occurs concurrently with a database operation like backup or restore. Retry your operation again when no other database-related operations are in progress.

Change manager messages

The messages in this section are issued by the Deployment Engine change manager component.

ACUCM3002W Change management operation *operation* failed for installable unit *installable_unit*.

Explanation: A failure occurred during the processing of an action that subsequently caused the change management operation to fail. This failed change management operation will usually cause the current change request to fail as well.

User response: This warning requires no response, but corrective actions could be necessary if the change request fails. Refer to the user response that is provided for any subsequent error messages.

ACUCM3004W Rollback of change management operation *operation* failed for installable unit *installable_unit*.

Explanation: Because a change request failed, Deployment Engine attempted to roll back each change management operation that was processed for the change request. The purpose of these rollbacks is to return the software to its original state. However, one of the rollbacks failed. Deployment Engine will not attempt to roll back any additional change management operations.

User response: This warning requires no response, but corrective actions will be necessary because the change request failed. Refer to the user response that is provided for any subsequent error messages.

ACUCM3005E Change request type *request_type* failed for software package *package_name*. Deployment Engine successfully rolled back the changes that it made when processing the change request.

Explanation: The change request failed because a change management operation failed. All changed software was restored to its original state.

User response: Resubmit the change request. If the problem persists, contact your support representative for assistance.

ACUCM3006E Change request type *request_type* failed for software package *package_name*. Rollback also failed; Deployment Engine was unable to restore the changes that it made when processing the change request.

Explanation: The change request failed because a change management operation failed. The changed software was not restored to its original state.

User response: Uninstall the software package and resubmit the change request. If the problem persists, contact your support representative for assistance.

Dependency checker messages

The messages in this section are issued by the Deployment Engine dependency checker component.

ACUDC0002W One or more dependencies were not satisfied for software package *package_name*.

Explanation: Prior to software deployment, Deployment Engine can check for different kinds of dependencies, including a computer's processing or resource capacity, the availability of other software, or a property value, relationship, or version associated with some particular software. In most cases an unsatisfied dependency will cause the current change request to fail.

User response: The software deployment program might issue a message that provides details about the dependencies that caused this problem or might prompt for additional information to address the problem. Contact your support representative if you need additional help to identify and fulfill the unsatisfied dependencies so that software deployment can go forward.

ACUDC0003W Deployment Engine found one or more integrity check violations in software package *package_name*.

Explanation: Prior to software installation, Deployment Engine makes sure that the software registered in the installation database can coexist with the software you are trying to install. Prior to software removal, Deployment Engine makes sure that no registered software depends on the software you are trying to remove. This function is referred to as integrity checking. Continuing when integrity check violations are present can adversely affect other software currently in use. Therefore an integrity check violation will usually cause the current change request to fail.

User response: The software deployment program might issue a message that provides details about the integrity check violation that caused this problem or might prompt for additional information to address the problem. Typically you do not want to install or remove any software as long as it continues to violate the Interoperability requirements of other software currently in use. Contact your support representative if you need additional help to 1) understand the integrity check violations or 2) make adjustments to the registered software so that software installation or removal can go forward.

Installation database messages

The messages in this section are issued by the deprecated Deployment Engine database backup command, **backupdb**, and the database restore command, **restoredb**.

ACUDB0006E The *command_name* command failed with return code *return_code*.

Explanation: The backupdb and restoredb commands are deprecated commands. For more information about the deprecated commands and their related return codes, refer to the database commands in *IBM Solution Install for Autonomic Computing: Developer's Guide and Reference*, Version 1.2.1 (provided with a previous release). Alternatively, you can use one of the new database commands documented elsewhere in this book.

User response: Take the appropriate corrective action based on the return code provided in this message, or retry your task using one of the new (nondeprecated) database commands documented in this book.

ACUDB0007E Deployment Engine could not find any database backup file in backup directory *backup_dir*. The database was not restored.

Explanation: Some typical reasons why a backup file cannot be found, from least to most typical, are: No backup was performed, the backup file was deleted, the backup file is in another location, or the backup file was specified using an incorrect file name. When a backup file cannot be found, Deployment Engine cannot restore the database.

User response: Make sure that a backup file is present either in the default location or in a location that you intend to specify using the *file_name* option in the database restore command. Then issue the command again. If you are specifying a backup file location, take care to indicate the correct location with a fully qualified file name and to verify that there are no typographical errors in the file path or name prior to issuing the command.

ACUDB0008W The database restore command completed successfully, but Deployment Engine was unable to restart the database service.

Explanation: When a database restore command is issued, Deployment Engine stops the database service before processing the command. This ensures that database information is not accessed or modified during command processing. When command processing completes, Deployment Engine restarts the database service. In this case, however, after the command completed successfully the database service indicated there was a problem when Deployment Engine attempted to restart the service.

User response: Further action might not be necessary. The database service might restart before or during the next Deployment Engine operation. If additional Deployment Engine operations fail with database errors, try using operating system tools to start the ACSI service, *acsisrv*, or try restarting the operating system. Then issue the database restore command again. If the problem persists, contact your support representative for assistance.

ACUDB0009E Deployment Engine was unable to shut down the database service. The database was not restored.

Explanation: When a database restore command is issued, Deployment Engine stops the database service before processing the command. This ensures that database information is not accessed or modified during command processing. When command processing completes, Deployment Engine restarts the database service. In this case, however, the database service indicated that there was a problem when Deployment Engine attempted to shut down the service. This might simply be a timing problem that was caused by the database service being busy with another task when Deployment Engine tried to shut it down.

User response: Issue the database restore command again. If the database service was previously busy and therefore could not shut down, the shutdown might be possible now. If the problem persists, contact your support representative for assistance.

ACUDB0010E The specified database backup file *file_name* already exists. To prevent overwriting any backup data currently in the file, Deployment Engine did not process the database backup command.

Explanation: To protect data currently in a backup file, Deployment Engine will not save database backup information to a file that already exists. Deployment Engine detected the presence of the file specified in the database backup command and therefore did not back up the database to that file.

ACUDB0011E • ACUDB0013E

User response: Move the specified database backup file to another location and issue the same command again. Alternatively, reissue your command with the fully qualified file name of a different database backup file.

ACUDB0011E Deployment Engine cannot create the specified database backup file *file_name*. The database was not backed up.

Explanation: A file name was specified as the backup file for storing the database backup information. The file name must be fully qualified, including both a valid file path and a file name, and you must have write permission in order for Deployment Engine to add the new file to the target directory. In this case, either the path for the specified database backup file does not exist, or you do not have the authority to write a new file to the target directory specified in the path. Therefore Deployment Engine cannot back up the database.

User response: Make sure you have the necessary write permission to create a new file in the target directory. Then reissue the command with a valid, fully qualified file name for the database backup file.

ACUDB0012E Deployment Engine cannot find or read the specified database backup file *file_name*. The database was not restored.

Explanation: A file name was specified as the backup file to retrieve the database backup information from. The file name must be fully qualified, including both a valid file path and a file name, and you must have read permission in order for Deployment Engine to access the backup information in the file. In this case, either the specified database backup file does not exist, or you do not have authority to read the file. Therefore Deployment Engine cannot restore the database.

User response: Make sure that the database backup file exists and you have the necessary permission to read the file. Then reissue the command with the valid, fully qualified file name of the database backup file.

ACUDB0013E The specified database backup file *file_name* does not contain database content. The database was not restored.

Explanation: A valid, fully qualified file name was specified for the file to retrieve the database backup information from. The specified file exists and you have the authority necessary to read the file. However, the file content does not consist of database backup information that is recognizable to Deployment Engine. Therefore Deployment Engine cannot restore the database. In this case, a nonbackup file might have been specified inadvertently.

User response: Verify the file name and location of your most recent database backup file. Then reissue the command, specifying the fully qualified file name of that database backup file.

Operating system touchpoint messages

The messages in this section are issued by the operating system touchpoint.

ACUOS0046W A failure occurred during the installation processing of action *action_name*, sequence number *seq_num*. Deployment Engine will ignore the failure and process the next action.

Explanation: This failure is usually associated with an authorization, disk space, or other operating system problem. Deployment Engine received instructions to ignore this failure and continue processing. As a result, this particular failure will not cause the change management operation to fail.

User response: This warning requires no response. However, corrective actions might be required at a later time to ensure the proper functioning of any related software or applications affected by the failed action.

ACUOS0059W A failure occurred during the undo processing of action *action_name*, sequence number *seq_num*. Deployment Engine will ignore the failure and process the next action.

Explanation: This failure is usually associated with an authorization, disk space, or other operating system problem. Deployment Engine received instructions to ignore this failure and continue processing. As a result, this particular failure will not cause the change management operation to fail.

User response: This warning requires no response. However, corrective actions might be required at a later time to ensure the proper functioning of any related software or applications affected by the failed action.

ACUOS0060W A failure occurred during the uninstallation processing of action *action_name*, sequence number *seq_num*. Deployment Engine will ignore the failure and process the next action.

Explanation: This failure is usually associated with an authorization, disk space, or other operating system problem. Deployment Engine received instructions to ignore this failure and continue processing. As a result, this particular failure will not cause the change management operation to fail.

User response: This warning requires no response. However, corrective actions might be required at a later time to ensure the proper functioning of any related software or applications affected by the failed action.

ACUOS0061W A failure occurred during the installation processing of action *action_name*, sequence number *seq_num*. Subsequent actions will not be processed.

Explanation: This failure is usually associated with an authorization, disk space, or other operating system problem. Deployment Engine received instructions to stop processing when this action failed. As a result, Deployment Engine will not process any more actions in this change management operation and the change management operation will probably fail.

User response: This warning requires no response, but corrective actions could be necessary at a later time if the change management operation and its associated change request fail. Refer to the user response that is provided for any subsequent error messages.

ACUOS0062W A failure occurred during the undo processing of action *action_name*, sequence number *seq_num*. Subsequent actions will not be processed.

Explanation: This failure is usually associated with an authorization, disk space, or other operating system problem. Deployment Engine received instructions to stop processing when this action failed. As a result, Deployment Engine will not process any more actions in this change management operation and the change management operation will probably fail.

User response: This warning requires no response, but corrective actions could be necessary at a later time if the change management operation and its associated change request fail. Refer to the user response that is provided for any subsequent error messages.

ACUOS0063W

ACUOS0063W A failure occurred during the uninstallation processing of action *action_name*, sequence number *seq_num*. Subsequent actions will not be processed.

Explanation: This failure is usually associated with an authorization, disk space, or other operating system problem. Deployment Engine received instructions to stop processing when this action failed. As a result, Deployment Engine will not process any more actions in this change management operation and the change management operation will probably fail.

User response: This warning requires no response, but corrective actions could be necessary at a later time if the change management operation and its associated change request fail. Refer to the user response that is provided for any subsequent error messages.

Chapter 11. Messages issued by commands

This chapter contains message help information for all of the error and warning messages issued by administrator commands of Deployment Engine that are referenced by a unique message reference number. This section also contains message help information for Deployment Engine messages common to all commands, and messages related to invoking the command line itself (main messages).

The messages are presented in the order listed below. The common and main messages are presented first. Then the messages for each Deployment Engine command are presented in alphanumeric order, beginning with the **de_backupdb** command.

- “Common messages” on page 76
- “Main messages” on page 88
- “de_backupdb messages” on page 95
- “de_fs messages” on page 97
- “de_help messages” on page 99
- “de_instmaint messages” on page 100
- “de_lsapp messages” on page 101
- “de_lsfeat messages” on page 102
- “de_lsmaint messages” on page 103
- “de_lsproc messages” on page 104
- “de_restoredb messages” on page 105
- “de_setdb messages” on page 107
- “de_uninstapp messages” on page 108
- “de_uninstfeat messages” on page 109
- “de_uninstmaint messages” on page 112
- “de_version messages” on page 113

Common messages

The messages in this section are common to multiple Deployment Engine administrator commands.

ACUEX0000E Deployment Engine could not load the logging level properties from the properties file *file_name*.

Explanation: An error occurred while trying to load some properties that govern the logging levels for messages and traces. Command processing continues using default values for the message and trace logging levels.

User response: See the trace log for more information about the cause of this error.

ACUEX0004E Deployment Engine cannot display the results of the query.

Explanation: Deployment Engine received an unexpected Java exception while trying to display the query results for the command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX0005E Deployment Engine cannot save the results of the query to a file.

Explanation: Deployment Engine received an unexpected Java exception while trying to save the query results for the command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX0006E The Deployment Engine version information could not be retrieved.

Explanation: An error occurred while querying the Deployment Engine installation database. Deployment Engine was unable to return the requested version information. The database might be temporarily inaccessible.

User response: The Deployment Engine installation database might have been busy with another operation, so try your command again. If the problem persists, check the message log for other messages associated with this one and follow the corrective actions provided in their user responses.

ACUEX0007E Deployment Engine could not load transform file *file_name*.

Explanation: Deployment Engine could not load the transform file that formats your query results. As a result, the command failed. The could-not-load problem described in the message can occur if the transform file was not installed with the Deployment Engine run-time environment.

User response: Contact the support personnel for your application to obtain assistance in locating and installing the transform file.

ACUEX0008E Deployment Engine could not format the query results.

Explanation: Deployment Engine received an unexpected Java exception when it tried to transform your query results to the requested format. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX0010E Deployment Engine could not display or save the query results after formatting them.

Explanation: Deployment Engine received an unexpected Java exception when it tried to display or save your query results. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX0018E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX0022E The maintenance specified in the command is not a valid software package.

Explanation: Deployment Engine attempted to apply or remove maintenance, in the form of a fix or incremental update software package, but there was a problem with the software package that caused your command to fail.

User response: This might be a problem with the file specification in the command, rather than with the software package itself. Make sure that the file you specified in the command is really a Deployment Engine-enabled software package. Also be sure that you specified the correct path and file name for the software package that you intended to apply or remove. If you specified the correct file but still received this message, then the software package might be corrupted or might violate Deployment Engine file packaging rules. In this case, you should contact the support personnel for your application to obtain assistance.

ACUEX0023E A descriptor in the software package for the maintenance specified in the command failed schema validation.

Explanation: Deployment Engine attempted to apply or remove maintenance, in the form of a fix or incremental update software package, but a descriptor in the software package was not valid. That is, the currently installed version of Deployment Engine could not validate the descriptor with the Deployment Engine schema. This problem is usually the result of a mismatch between the version of Deployment Engine that is performing the validation and the version of the software package that is being validated.

User response: Make sure that the software package for the maintenance you are working with is intended to be used with the version of Deployment Engine that you are currently running on the computer. That is, make sure that the software package is at the same version (or an earlier version) as Deployment Engine. Once these versions are made compatible, you can issue your command again. If this failure occurs and the versions are already compatible, then the descriptor probably has internal, schema-related problems. In this case, you should contact the support personnel for your application to obtain assistance.

ACUEX0024E The command failed because Deployment Engine is busy. Please try again later.

Explanation: The command that you issued requires Deployment Engine to access its installation database. Deployment Engine must process database operations singly to prevent the corruption of its database. Because another operation was already in progress, the new operation initiated by your command could not be processed.

User response: Make sure you are not doing multiple operations that affect the database at the same time. Examples of multiple operations include simultaneous software deployments and a software deployment that occurs concurrently with a database operation like backup or restore. Retry your command again when no other database-related operations are in progress.

ACUEX0026E The command failed due to an unknown cause.

Explanation: An unexpected error was received from the Deployment Engine change manager component. The cause of this error is not known.

ACUEX0028E • ACUEX0033E

User response: Check the message log for other messages associated with this one and follow the corrective actions provided in their user responses.

ACUEX0028E You do not have the required permission to perform this operation. Ask your system administrator if you can be granted the necessary authorization.

Explanation: A Deployment Engine operation failed because you are not authorized to access a required file.

User response: Check the message log for additional details about this authorization error. Then contact your system administrator and provide this information when requesting authorization.

ACUEX0029E Update not found: *version_number*.

Explanation: Deployment Engine was attempting to uninstall an incremental update from an application, but could not find an update with the version number specified in your command. Either the application does not contain that particular incremental update, or the wrong version was specified. The specified version number is indicated in the message.

User response: Make sure that an incremental update with the version number indicated in the message is currently applied to the application. If the correct incremental update is found but has a different version number, and you still want to uninstall it, reissue your command with the correct version number.

ACUEX0030E Feature not found: *feature_id*.

Explanation: Deployment Engine was attempting to uninstall a feature from an application, but could not find a feature with the feature identifier specified in your command. Either the application does not contain that particular feature, or the wrong feature identifier was specified. The specified feature identifier is indicated in the message.

User response: Make sure that a feature with the feature identifier indicated in the message is currently part of the application. If the correct feature is found but has a different feature identifier, and you still want to uninstall it, reissue the command with the correct feature identifier.

ACUEX0031E Fix not found: *fix_name*.

Explanation: Deployment Engine was attempting to uninstall a fix from an application, but could not find a fix with the fix name specified in your command. Either the application does not contain that particular fix, or the wrong fix name was specified. The specified fix name is indicated in the message.

User response: Make sure that a fix with the fix name indicated in the message is currently part of the application. If the correct fix is found but has a different fix name, and you still want to uninstall it, reissue the command with the correct fix name.

ACUEX0033E The application specified in the command cannot be uninstalled because the following installed applications or application components depend on it: *UUIDs*

Explanation: The command failed because another application or application component depends on the application that you want to uninstall. To protect that dependent software from problems that might occur if the application you specified was removed, Deployment Engine has failed the uninstallation command and returned this message. The UUIDs of the dependent software are identified in the message. They can be used in the process of determining what dependent software is blocking your uninstallation command.

User response: You will probably need assistance from your application's support personnel to determine what dependent software is preventing your uninstallation from going forward. However, if you already know which software on your computer depends on the application you want to uninstall, and you are sure you no longer require that software, you can remove it using the **de_uninstall** command and then reissue your original command.

Alternatively, you can use the **de_lsapp** command to try to identify which applications are associated with the dependent UUIDs named in the message. If the **de_lsapp** command returns one application instance only, this is the application instance that is blocking your command. To proceed, you need to remove the blocking application instance. If the **de_lsapp** command returns multiple application instances, this means that the dependent application is one of these returned application instances, but you cannot be certain which one. That is, the dependent application could not be identified by its UUID alone (you usually need a UUID and a discriminant to identify an application uniquely). In this case, you will need outside assistance to make a determination. If the **de_lsapp**

command returns no information, the UUID probably belongs to an application component. It therefore cannot be identified by `de_lsapp`, which only returns application information, and again you will need outside assistance.

As a last resort, you can use the `-force` option. Reissuing your command with the `-force` option will allow application uninstallation to go forward, regardless of any other software that depends on the application. Using the `-force` option usually causes other difficulties later (for example, the dependent application might no longer function properly), and therefore a force is not recommended.

ACUEX0034E An error occurred when backing up or restoring the Deployment Engine installation database. The backup file is corrupted.

Explanation: The backup or restore action failed because the backup file is corrupt.

User response: Check the message log for additional details about this error.

ACUEX0035E The requested backup file is not valid.

Explanation: The restore action failed because the backup file is not valid.

User response: Check the message log for additional details about this error.

ACUEX0036E You do not have permission to access this backup file.

Explanation: The backup or restore action failed because you do not have the required authorization to the backup file.

User response: Request authorization from your system administrator to access this backup file.

ACUEX0038E The Deployment Engine run-time environment is not installed.

Explanation: Deployment Engine is not installed on the computer, or cannot be found.

User response: Contact the support personnel for your application to obtain assistance.

ACUEX0039E The backup file already exists.

Explanation: The backup file specified already exists and cannot be replaced.

User response: Specify a different backup file name.

ACUEX0040E The backup file path specified is not valid.

Explanation: The backup file path specified is not a valid file path name.

User response: Specify a valid backup file name.

ACUEX0041E You have attempted an unauthorized installation of the Deployment Engine run-time environment.

Explanation: The installation of the Deployment Engine run-time environment has failed. You might have tried to install Deployment Engine into a directory that you are not currently authorized to access.

User response: Make sure you are authorized to access the target directory. If you are not authorized, either obtain authorization from a system administrator to access the directory that you are installing into, or install Deployment Engine into another directory that you do have access to.

ACUEX0042E The specified database port is not valid.

Explanation: The port number specified cannot be used to access the Deployment Engine database service.

User response: Use a valid port number for the database service.

ACUEX0043E Deployment Engine has returned the following error: *exception_message*.

Explanation: An unexpected error was received during a Deployment Engine operation.

User response: See the message log for previous errors that caused this message.

ACUEX0044E The requested backup file cannot be found.

Explanation: The backup file specified cannot be found.

User response: Specify a different backup file name.

ACUEX0054E Exception, Stack:

Explanation: An unexpected Java exception was received from a Deployment Engine API. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: If the cause of the error cannot be determined from the details of the exception, contact the support personnel for your application to obtain assistance.

ACUEX0055E An error occurred when attempting to schedule an Deployment Engine database backup.
See the log file *file_name* for details.

Explanation: An unexpected Java exception was received from a Deployment Engine API.

User response: See the message log for previous errors that caused this message.

ACUEX0056E An error occurred when attempting to schedule a file scan.
See the log file *file_name* for details.

Explanation: An unexpected Java exception was received from a Deployment Engine API.

User response: See the message log for previous errors that caused this message.

ACUEX0057E An error occurred when attempting to remove the scheduled Deployment Engine database backup.
See the log file *file_name* for details.

Explanation: An unexpected Java exception was received from a Deployment Engine API.

User response: See the message log for previous errors that caused this message.

ACUEX0058E An error occurred when attempting to remove the scheduled file scan.
See the log file *file_name* for details.

Explanation: An unexpected Java exception was received from a Deployment Engine API.

User response: See the message log for previous errors that caused this message.

ACUEX0061E An error occurred when attempting to list the scheduled processes.
See the log file *file_name* for details.

Explanation: An unexpected Java exception was received from a Deployment Engine API.

User response: See the message log for previous errors that caused this message.

ACUEX0073E An error has occurred when backing up or restoring the Deployment Engine installation database.
Deployment Engine is in use.

Explanation: The Deployment Engine installation database is currently in use by another process on the computer.

User response: Wait until all other Deployment Engine operations on the computer are complete and try the command again.

ACUEX0074E The uninstall operation failed because of a Deployment Engine database error. The ACSI database service may not be started.

Explanation: While running an uninstall operation, the Deployment Engine database service could not be reached.

User response: Ensure that the ACSI service is running on the computer before running this command.

ACUEX0075E A general Deployment Engine error has occurred.

Explanation: An unexpected Java exception was received from a Deployment Engine API.

User response: See the message log for previous errors that caused this message.

ACUEX0079E The database query failed because of a Deployment Engine database error. The ACSI database service may not be started.

Explanation: While querying the Deployment Engine installation database, Deployment Engine could not reach the ACSI database service.

User response: Ensure that the ACSI service is running on the computer before reissuing this command.

ACUEX0080E Software package *file_name* cannot be uninstalled.

Explanation: Deployment Engine found an installed full update that matched the software package specified in the command. However, a full update cannot be uninstalled. When the full update was originally deployed, it either completely replaced an existing application or it performed a fresh installation of an application that did not previously exist. The only way to remove a full update is to uninstall the entire application--which, in fact, is the full update.

User response: If you really want to remove the complete contents of the full update, you must remove the application itself by using the **de_uninstapp** command. Refer to the *Autonomic Deployment Engine for Administrators* book for information about how to use this command.

ACUEX0081E No installed maintenance matches the software package *file_name*

Explanation: No installed instance of the specified software package was found.

User response: Specify a valid maintenance package.

ACUEX0082E The following maintenance packages need to be installed separately:
file_list

Explanation: A request was made to install multiple incremental updates to an application. Incremental updates must be installed one at a time.

User response: Specify only one incremental update to be installed.

ACUEX0083E Deployment Engine only supports installation of multiple fixes for one application. All fix software packages must have the same UUID.

Explanation: The fix software packages specified apply to different applications. Maintenance may be installed to only one application at a time.

User response: Apply maintenance to one application at a time.

ACUEX0084E The fix contained in software package
filename_1
has the same fix name as the fix contained in software package
filename_2

Explanation: Two fix packages specified contain a fix with the same name. Each fix software package must contain a unique fix.

User response: Specify a list of unique fix software packages on the command.

ACUEX0085E • ACUEX0093E

ACUEX0085E No valid maintenance packages found.

Explanation: None of the maintenance packages specified contained valid maintenance.

User response: Specify one or more valid maintenance packages.

ACUEX0086E No file name entered for maintenance package.

Explanation: The command was specified with a blank maintenance package name.

User response: Specify one or more valid maintenance packages.

ACUEX0088E Software package *file_name* is an unsupported maintenance package.

Explanation: The type of maintenance contained in the specified software package is not supported.

User response: Specify one or more valid maintenance packages.

ACUEX0089E The maintenance selected cannot be installed to any application on the computer.

Explanation: The maintenance package specified on the command does not apply to any application installed on the computer.

User response: Specify one or more valid maintenance packages.

ACUEX0090E More than one application matches. Please provide more input to select a specific application.

The following discriminants are installed:

discriminant_list

Explanation: The maintenance specified could apply to more than one installed application instance.

User response: Use the **-uuid** and **-disc** options of the command to specify a unique application instance.

ACUEX0091E Exception, Stack:

Explanation: An unexpected Java exception was received while installing maintenance. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: If the cause of the error cannot be determined from the details of the exception, contact the support personnel for your application to obtain assistance.

ACUEX0092E Exception, Stack:

Explanation: An unexpected Java exception was received while uninstalling maintenance. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: If the cause of the error cannot be determined from the details of the exception, contact the support personnel for your application to obtain assistance.

ACUEX0093E The maintenance selected cannot be installed to any application matching the discriminant:

discriminant

Explanation: For the maintenance selected, no installed application instance can be found with the discriminant specified.

User response: Specify a discriminant matching an installed instance of the application for which the maintenance is intended.

ACUEX0095E Exception, Stack:

Explanation: An unexpected Java exception was received during a Deployment Engine change operation. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: If the cause of the error cannot be determined from the details of the exception, contact the support personnel for your application to obtain assistance.

ACUEX0099E Change failed.

Explanation: A Deployment Engine change operation failed.

User response: See the message log for previous errors that caused this message.

ACUEX0100E Simultaneous change attempt.

Explanation: An attempt was made to begin one Deployment Engine change management operation while another one was still underway.

User response: Ensure that no other Deployment Engine commands are active and try the command again.

ACUEX0102E Lock not allowed.

Explanation: An attempt was made to begin a Deployment Engine change management operation, but another process on the computer has an active write lock on the IU Registry.

User response: Ensure that no other Deployment Engine commands are active and try the command again.

ACUEX0104E General exception.

Explanation: An unexpected Java exception was received while performing a change management operation.

User response: See the message log for previous errors that caused this message.

ACUEX0105E System check failure.

Explanation: During a change management operation, the attempted operation failed dependency checking or integrity checking.

User response: See the message log for previous errors that caused this message.

ACUEX0106E This fix is not backward compatible with the current application.

Explanation: The selected maintenance failed a dependency check. The fix cannot be applied without breaking the dependency of other applications on the computer.

User response: Check the version of the application currently installed, and the versions required by other applications on the computer.

ACUEX0107W None of the features to which the fix can be applied are currently installed.

Explanation: The maintenance selected failed a dependency check. The failing check is the <featureHasChildren> constraint. There are no applicable features.

User response: Check the documentation that came with the maintenance package to ensure that it applies to your environment.

ACUEX0108E This fix has been installed as many times as the computer will allow.

Explanation: The maintenance selected failed a dependency check. The failing check is the <maximumInstances> constraint.

ACUEX0109E • ACUEX0115E

User response: Check the documentation that came with the maintenance package to ensure that it applies to your environment.

ACUEX0109E This incremental update cannot be applied to the selected application. It is either the wrong application or the wrong version.

Explanation: The maintenance selected failed a dependency check. There are no instances of the application installed to which the update can be applied.

User response: Check the documentation that came with the maintenance package to ensure that it applies to your environment.

ACUEX0110E The computer does not currently satisfy all the requirements necessary to apply the fix.

Explanation: The maintenance selected failed a dependency check. The failing check is the <requirements> section of the IU deployment descriptor.

User response: Check the documentation that came with the maintenance package to ensure that it applies to your environment.

ACUEX0111E This fix software package has no content available to be installed.

Explanation: The maintenance selected failed a dependency check. The failing check is the <rootHasChildren> constraint.

User response: Check the documentation that came with the maintenance package to ensure that it applies to your environment.

ACUEX0112E This fix software package cannot be applied since it does not supersede all required fixes:

Explanation: The maintenance selected failed a dependency check. The failing check is the superseded fixes check. There are fixes installed to the application which are not labeled as being superseded in this software package.

User response: To install this software package, one of two things can be done.

- All the non-superseded fixes can be removed to allow the install of this software package.
 - Find the fix software package that is both superseded by this software package and that supersedes all of the currently installed fixes on the application.
-

ACUEX0113E This fix software package cannot be applied in the current topology. This package may have been built for a different operating system.

Explanation: The maintenance selected failed a dependency check. The failing check is the <topology> section of the IU deployment descriptor.

User response: Check the documentation that came with the maintenance package to ensure that it applies to your environment.

ACUEX0114E The application is in a state that does not allow you to apply the new maintenance.

Explanation: Before you can apply the new maintenance to the application, you first need to reapply one or more previous incremental updates or fixes. This situation can occur when an incremental update or fix is applied to a base software package, then one or more features are later added that generate an UpdatesNeedsToBeReapplied warning.

User response: Reapply the required previous incremental updates or fixes before attempting to apply the new maintenance.

ACUEX0115E There is a dependency that has not been met:

Explanation: A change management operation attempted to update an application with a fix containing dependencies that have not been met.

User response: See the message log for details on the dependencies that caused this message.

ACUEX0116E This software package cannot be applied since it does not supersede all required fixes.

Explanation: A change management operation attempted to update an application that has fixes applied to it, and the fixes are not superseded by the updating software package.

User response: To install this software package, one of two things can be done.

- All the nonsuperseded fixes can be removed to allow the installation of this software package.
- Find the fix software package that is both superseded by this software package and that supersedes all of the currently installed fixes on the application.

ACUEX0117E This fix has been superseded by another software package on the computer:

Explanation: The change management operation attempted to update an application with a fix that has been superseded.

User response: Ensure that you have the latest maintenance packages for the application, and that fixes are being applied in the correct order.

ACUEX0118E This fix cannot be applied due to a dependency on another application.

Explanation: Another application using the application targeted in the change management operation will be impacted if this fix is applied.

User response: To ignore the impact on the other application, use the **-force** option when applying this fix.

ACUEX0119E This fix conflicts with another fix on the computer and cannot be applied.

Explanation: A different fix for the application has an prerequisite (XREQ) relationship with this fix. The two fixes cannot be simultaneously applied.

User response: Remove the conflicting fix before applying this fix.

ACUEX0120E A Deployment Engine change operation failed. The causes for the failure follow:

failure_list

Explanation: An unexpected error occurred while Deployment Engine was performing a change operation on the computer. The causes for the failure are included in this message.

User response: See the message log for other errors that caused this message.

ACUEX0121E Requirement failure:

Explanation: One or more requirements were not met while attempting to perform a change operation on the computer.

User response: See the message log for other errors that caused this message.

ACUEX0122E The requested update version is not valid.

Explanation: The current version of the application is not the update version specified to be removed, or the update cannot be removed because it is not an incremental update.

User response: Specify an update version that can be removed.

ACUEX0123E One or more requested fix names cannot be found:

fix_name_list

Explanation: No maintenance that matches the fix names specified in the message is installed for the requested application.

User response: Specify one or more installed fixes to be uninstalled.

ACUEX0125E • ACUEX0132E

ACUEX0125E Dependency checking failed:

Explanation: One or more dependency checks have failed while processing the requested command.

User response: See the message log for more information about the error.

ACUEX0126E Integrity checking failed:

Explanation: One or more integrity checks have failed while processing the requested command.

User response: See the message log for more information about the error.

ACUEX0127E The maintenance you specified cannot be uninstalled. The following maintenance must first be removed:

Explanation: The maintenance specified in your command failed a dependency check. There are fixes, updates, or both that are currently applied to the application, and they must be removed before the specified maintenance can be uninstalled.

User response: Before you can uninstall the maintenance that you specified, you must first do one of the following things:

- Manually remove the fixes or updates that caused the error.
- Issue the command with the **-auto** option to remove the fixes or updates that caused the error.

Then reissue your original command to uninstall the maintenance.

ACUEX0128E This operation requires a prerequisite software package not found on this computer.

Explanation: One or more required software packages are not installed on the computer. The requirements verification failed.

User response: See the message log for more detailed information about the error.

ACUEX0129E Either no installed maintenance package matched the specified software package, or this application might have superseded fixes. Retry using the -auto option.

Explanation: There are no installed maintenance packages that match the maintenance package that you specified. If you specified a fix, it is possible that the fix has been superseded and is no longer visible.

User response: See the message log for more information about the error.

ACUEX0130E The following requirements failed:

Explanation: During system verification, one or more requirements failed.

User response: See the message log for more information about the error.

ACUEX0131E CheckID:

Explanation: During system verification, one or more checks failed.

User response: See the message log for more information about the error.

ACUEX0132E Name:

Explanation: During system verification, one or more requirements failed.

User response: See the message log for more information about the error.

ACUEX0133E The following maintenance packages need to be uninstalled separately:

file_name_list

Explanation: A request was made to uninstall multiple incremental updates to an application. Incremental updates must be uninstalled one at a time.

User response: In your command, specify only one incremental update to be uninstalled, beginning with the most recent update.

ACUEX0134E No installed maintenance package matched the software package requested.

Explanation: There are no installed maintenance packages that match the software package specified in the command.

User response: See the message log for more information about the error.

ACUEX0135E There was an error controlling the database service.

Explanation: The database cannot be started or stopped at this time.

User response: See the message log for more information about the error.

Main messages

The messages in this section are messages that you might get during invocation of the command line.

ACUEX1000E You must type a Deployment Engine command.

Explanation: The Deployment Engine main program was called but no command name was specified.

User response: Invoke Deployment Engine administration commands by using one of the documented command files, usually a UNIX shell script or a Windows .cmd file.

ACUEX1001E Unknown input data.

Explanation: The Deployment Engine main program was called but unrecognizable data were passed as arguments.

User response: Invoke Deployment Engine administration commands by using one of the documented command files, usually a UNIX shell script or a Windows .cmd file.

ACUEX1002E A severe internal error has occurred. `ArrayIndexOutOfBoundsException`.

Explanation: An unexpected Java exception was received while processing an administration command.

User response: Contact the support personnel for your application to obtain assistance.

ACUEX1003E `ArrayIndexOutOfBoundsException`, Stack:

Explanation: An unexpected Java exception was received while processing an administration command. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: Contact the support personnel for your application to obtain assistance.

ACUEX1004E Command not found: *command_name*

Explanation: The Deployment Engine main program was called but the command name passed to the program was not found as a plug-in.

User response: Invoke Deployment Engine administration commands by using one of the documented command files, usually a UNIX shell script or a Windows .cmd file.

ACUEX1005E A severe internal error has occurred. `CoreException`.

Explanation: An unexpected Eclipse exception was received while processing an administration command.

User response: See the message log for more information about the error.

ACUEX1006E `CoreException`, Stack:

Explanation: An unexpected Eclipse exception was received while processing an administration command. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: If the cause of the error cannot be determined from the details of the exception, contact the support personnel for your application to obtain assistance.

ACUEX1007E There are no Deployment Engine plug-ins.

Explanation: The Deployment Engine main program was called but it could not find any command plug-ins installed.

User response: See the message log for more information about the error.

ACUEX1008E The command could not be run. `ExceptionInInitializerError`.

Explanation: An unexpected Java exception was received while processing an administration command. This error is caused by a problem initializing a Deployment Engine command plug-in.

User response: See the message log for more information about the error.

ACUEX1009E `ExceptionInInitializerError`, `Stack`:

Explanation: An unexpected Java exception was received while processing an administration command. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: If the cause of the error cannot be determined from the details of the exception, contact the support personnel for your application to obtain assistance.

ACUEX1010E `Exception`, `Stack`:

Explanation: An unexpected Java exception was received while processing an administration command. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: If the cause of the error cannot be determined from the details of the exception, contact the support personnel for your application to obtain assistance.

ACUEX1011E A value is required for option *option_name*.

Explanation: The option requires a value and has no default value.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1012E Option *option_name* does not require a value.

Explanation: This option is a keyword option and no value can be specified.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1013E The validator rule file contains data that is not valid. Cannot convert number:
`NumberFormatException`, `Stack`:

Explanation: A Deployment Engine administration command failed because of an internal error while parsing the command line. The internal syntax file for the command contains a numeric value that is not valid. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: Contact the support personnel for your application to obtain assistance.

ACUEX1014E The validator rule file contains data that is not valid. Cannot convert number:
`NumberFormatException`. The value *value* is not valid.

Explanation: A Deployment Engine administration command failed because of an internal error while parsing the command line. The internal syntax file for the command contains a numeric value that is not valid. This message includes the value that caused the error.

User response: Contact the support personnel for your application to obtain assistance.

ACUEX1015E Required data for option *option_name* is missing or not valid.

Explanation: The value specified for the option named in the message is not valid.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1016E There are more values than allowed for option *option_name*.

Explanation: Too many values were supplied for the option.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1017E The integer value specified was less than the minimum value allowed, *value*.

Explanation: The numeric value specified for the option is less than the minimum allowed by the command.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1018E The integer value specified was greater than the maximum allowed, *value*.

Explanation: The numeric value specified for the option is greater than the maximum allowed by the command.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1019E The validator rule file contains data that is not valid. Cannot convert number:
NumberFormatException. The minimum value is *value*. The maximum value is *value*.

Explanation: A Deployment Engine administration command failed because of an internal error while parsing the command line. The internal syntax file for the command contains a numeric value that is not valid. This message includes the value that caused the error.

User response: Contact the support personnel for your application to obtain assistance.

ACUEX1020E A severe internal error occurred.

Explanation: An unexpected Java exception was received while processing an administration command.

User response: See the message log for more information about the error.

ACUEX1021E The string size is less than the minimum allowed, *string_size*.

Explanation: The value of the option is shorter than the minimum length allowed.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1022E The string size is greater than the maximum allowed, *string_size*.

Explanation: The value of the option is longer than the maximum length allowed.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1023E Cannot validate time. NullPointerException. Stack:

Explanation: A date or time value specified as a option in a command is missing or not valid.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1024E Cannot validate time because the date or time for the value *date_time* is either null or not valid.

Explanation: A date or time value specified as an option in a command is missing or not valid.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1025E The validator rule file cannot be null.

Explanation: An internal error occurred while processing an administration command. The name of the internal syntax file for the command is null.

User response: Contact the support personnel for your application to obtain assistance.

ACUEX1029E Cannot verify file *file_name*.

Explanation: The value of the option must be a valid file or directory, but the file name could not be verified.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1030E Cannot validate host. Malformed URL Exception. Stack:

Explanation: The command option must specify the URL of a host, but the value specified is not a valid URL.

User response: Correct the syntax of the option value to be a valid URL.

ACUEX1031E Cannot validate host. Malformed URL Exception. The URL *url* is not valid.

Explanation: The command option must specify the URL of a host, but the value specified is not a valid URL.

User response: Correct the syntax of the option value to be a valid URL.

ACUEX1032E Cannot validate host. IOException. Stack:

Explanation: The command option must specify the URL of a host, but a connection could not be opened to the URL supplied. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: Check the URL value and ensure that a connection can be established.

ACUEX1033E Cannot validate host: IOException. Cannot connect to URL *url*.

Explanation: The command option must specify the URL of a host, but a connection could not be opened to the URL supplied.

User response: See the message log for more information about the error.

ACUEX1034E Failed input option validation:

Explanation: The syntax of the command options is incorrect. Other messages are issued which describe the syntax error in more detail.

User response: See the message log for the related error messages.

ACUEX1035E Found duplicate option value *option_value*.

Explanation: The same value was specified for an option multiple times. Only one occurrence of the same option value is allowed for the option.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1036E Found duplicate option *option_name*.

Explanation: The same option was specified more than once in the command. Only one occurrence of the option is allowed.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1037E Found unknown option *option_name*.

Explanation: The option name specified is not a valid option for the command.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1038E One or more required options were missing from administration command *command_name*.

Explanation: One or more options that are required for this administration command were not specified.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1039E The option *option_name* **is not allowed.**

Explanation: The option that you specified is not allowed in combination with another option you specified.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1040E A dependent option, *option_name*, **is missing from the command.**

Explanation: An option that you entered in the command requires that the indicated option is also specified.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1041E Found an incompatible option *option_name*.

Explanation: The option entered in the command is not allowed with another option you specified.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1042E Cannot read command syntax file. IOException. The file *file_name* **cannot be read.**

Explanation: An unexpected Java IOException was received while reading the internal syntax file for the command. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: Contact the support personnel for your application to obtain assistance.

ACUEX1043E Cannot parse command syntax file. SAXException. The file *file_name* cannot be parsed.

Explanation: An unexpected Java SAXException was received while reading the internal syntax file for the command. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: Contact your the support personnel for your application to obtain assistance.

ACUEX1044E Cannot parse command syntax file. ParserConfigurationException. The file *file_name* cannot be parsed.

Explanation: An unexpected Java ParserConfigurationException was received while reading the internal syntax file for the command. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: Contact your the support personnel for your application to obtain assistance.

ACUEX1045E An error occurred while parsing the command syntax file. Qualified name not found: *qualified_name*

Explanation: An error occurred while parsing the internal syntax file for the command.

User response: Contact the support personnel for your application to obtain assistance.

ACUEX1046E An error occurred while parsing the command syntax file. The value *value* is not valid.

Explanation: An error occurred while parsing the internal syntax file for the command.

User response: Contact the support personnel for your application to obtain assistance.

ACUEX1047E Cannot parse command syntax file. SAXParseException. Stack:

Explanation: An unexpected Java SAXParseException was received while processing an administration command. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: If the cause of the error cannot be determined from the details of the exception, contact the support personnel for your application to obtain assistance.

ACUEX1051E Either more values were specified than option *option_name* allows, or the option does not require any values.

Explanation: The command option was specified with too many values.

User response: To display syntax and option information for this command, issue your command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1052E The file *file_name* is not valid.

Explanation: The name of the file indicated in the message is not a valid file name.

User response: Enter a file name which conforms with file naming rules for your operating system.

ACUEX1053E The specified name *file_name* cannot refer to a directory.

Explanation: The command requires that you specify a file name. However, a directory already exists with the same name as the file name that you specified. A directory name is not allowed.

User response: Choose a different file name, rename the directory, or remove the directory.

ACUEX1054E Access to privileged file *file_name* is denied. You might not have the required permissions.

Explanation: Your user ID was not able to read, write, create, or delete the file indicated in the message.

User response: See the message log for more information about the error. If access to the file is needed but currently is not allowed, contact your system administrator and request authorization.

ACUEX1055E The specified date/time value *date_time* is not valid.

The date/time value must adhere to one of the following formats:

YYYY-MM-DDThh:mm:ss

hh:mm:ss (current date is assumed)

An example of a valid date/time value for December 25th, 2005 at 7:12 PM:

2005-12-25T19:12:00

An example of a valid date/time value for today at 8:30 AM:

08:30:00

Explanation: The date/time value that you specified as a command option is not valid.

User response: Reissue your command using a date/time value that complies with the format information provided in the message.

ACUEX1056E Administration commands cannot be processed simultaneously. Please wait until Deployment Engine is no longer in use to issue a command.

Explanation: When an administration command runs, it creates a temporary lock file named <user.home>/IBM/acuex/workspace/.metadata/.plugins/com.ibm.ci.six.cli.main/.deex_lock. When you attempted your command, Deployment Engine detected the temporary lock file and assumed that another command was already running.

User response: Make sure that no administration commands are currently running on the computer. If your command issues this message incorrectly, delete the temporary lock file and try the command again.

ACUEX1058W Failed to get lock file. Lock ignored.

Explanation: An unexpected Java exception was received while attempting to create a lock file. This error does not prevent the administration command from running.

User response: If the error continues to occur on every command, contact the support personnel for your application to obtain assistance.

ACUEX1060E A severe error occurred while processing the administration command *command_name*.

Explanation: An unexpected Java exception was received while processing the command named in the message.

User response: See the message log for more information about the error.

ACUEX1061E A severe error occurred while processing an administration command. Stack:

Explanation: An unexpected Java exception was received while processing an administration command. This message is logged with the details of the exception, including a stack trace which shows the sequence of program calls invoked before the exception occurred.

User response: If the cause of the error cannot be determined from the details of the exception, contact the support personnel for your application to obtain assistance.

de_backupdb messages

The messages in this section are issued by the Deployment Engine **de_backupdb** command.

ACUEX1105E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX1106E Deployment Engine could not process the **de_backupdb** command.

Explanation: Deployment Engine received an unexpected Java exception while processing the **de_backupdb** command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX1107E Deployment Engine could not overwrite backup file *file_name*. This file is being used by another program.

Explanation: The backup file named in the message does exist, but it could not be deleted because it is currently in use.

User response: Make sure that the backup file is not open or locked by another program. Once the file is free, try your command again.

ACUEX1109E The name specified for the backup file, *file_name*, is a directory name.

Explanation: The backup file name that you specified in your **de_backupdb** command is the name of a directory on the computer. The name of the backup file cannot be the same as the name of an existing directory.

User response: Reissue your command with a unique backup file name that does not match the name of a directory on the computer.

ACUEX1113E The value specified for the **-time** option, *date_time*, is not valid.

The value for the **-time** option must be in one of the following formats:

<yyyy-mm-dd>T<hh:mm:ss>

<hh:mm:ss> (assumes that the backup is to occur on the current date)

An example of a valid specification for December 25th, 2006 at 7:12 p.m.:

2006-12-25T19:12:00

An example of a valid specification for today at 8:30 a.m.:

08:30:00

Explanation: The value you specified for the **-time** option is not in the correct format.

User response: See the text of this message for examples of the correct date and time format. To display syntax and option information for this command, issue the command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1114E The time that you specified, *date_time*, is earlier than the current time, *date_time*.

Explanation: The value you specified for the **-time** option is earlier than the current time. A database backup cannot be scheduled for a time that has already passed.

User response: Reissue your command with a time value for the database backup that is later than the current time.

ACUEX1115E You do not have the required permission to perform this operation. Ask your system administrator if you can be granted the necessary authorization.

Explanation: The user mode of the Deployment Engine run-time environment that you are using determines the permissions you need to issue this command. For a multiuser mode Deployment Engine run-time environment, only a root user can issue this command. For a single-user mode Deployment Engine run-time environment, a nonroot user can issue this command.

Also, the use of some of the options for this command are restricted. The **-time**, **-freq**, and **-unschd** options are only available in multiuser mode. (You can display the option information for this command by issuing the command with the **-help** option only.)

User response: If you do not know whether your copy of Deployment Engine was installed in multiuser mode or single-user mode, you can find out by issuing the **de_version** command. The **de_version** command displays the Deployment Engine user mode as well as the version.

For more details about command authorization, refer to the chapter on working with commands in the *Autonomic Deployment Engine for Administrators* book.

After determining the permissions required, contact your system administrator to request any additional authorization needed to run this command.

de_fs messages

The messages in this section are issued by the Deployment Engine `de_fs` command.

ACUEX1155E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX1156E Deployment Engine could not process the `de_fs` command.

Explanation: Deployment Engine received an unexpected Java exception while processing the `de_fs` command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX1158E The value specified for the `-time` option, `date_time`, is not valid.

The value for the `-time` option must be in one of the following formats:

`<yyyy-mm-dd>T<hh:mm:ss>`

`<hh:mm:ss>` (assumes that the file scan is to occur on the current date)

An example of a valid specification for December 25th, 2006 at 7:12 p.m.:

`2006-12-25T19:12:00`

An example of a valid specification for today at 8:30 a.m.:

`08:30:00`

Explanation: The value you specified for the `-time` option is not in the correct format.

User response: See the text of this message for examples of the correct date and time format. To display syntax and option information for this command, issue the command with the `-help` option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1159E The time that you specified, `date_time`, is earlier than the current time, `date_time`.

Explanation: The value you specified for the `-time` option is earlier than the current time. A file scan cannot be scheduled for a time that has already passed.

User response: Reissue your command with a time value for the file scan that is later than the current time.

ACUEX1160E You do not have the required permission to perform this operation. Ask your system administrator if you can be granted the necessary authorization.

Explanation: The user mode of the Deployment Engine run-time environment that you are using determines the permissions you need to issue this command. A multiuser mode Deployment Engine run-time environment is required to use this command. Therefore you must be a root user for your computer operating system in order to issue this command.

User response: Make sure you are using a multiuser mode Deployment Engine run-time environment. If you do not know whether your copy of Deployment Engine was installed in multiuser mode or single-user mode, you can find

out by issuing the **de_version** command. The **de_version** command displays the Deployment Engine user mode as well as the version.

For more details about command authorization, refer to the chapter on working with commands in the *Autonomic Deployment Engine for Administrators* book.

After determining the root user permissions required for your operating system, contact your system administrator to request any additional authorization needed to run this command.

de_help messages

The messages in this section are issued by the Deployment Engine **de_help** command.

ACUEX1205E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX1206E Deployment Engine could not process the **de_help** command.

Explanation: Deployment Engine received an unexpected Java exception while processing the **de_help** command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

de_instmaint messages

The messages in this section are issued by the Deployment Engine **de_instmaint** command.

ACUEX1262E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX1263E Deployment Engine could not process the **de_instmaint** command.

Explanation: Deployment Engine received an unexpected Java exception while processing the **de_instmaint** command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX1265E Deployment Engine returned the following message:

de_msg

Explanation: An unexpected Deployment Engine message was received while processing the **de_instmaint** command.

User response: See the message log for more information about the error.

ACUEX1266E *de_msg*

Explanation: An unexpected Deployment Engine error was received while processing the **de_instmaint** command. This message shows you the text of the message associated with that Deployment Engine error.

User response: See the message log for more information about the error.

ACUEX1268W Request failed to complete.

Explanation: The **de_instmaint** command failed.

User response: See the message log for more information about the error.

de_lsapp messages

The messages in this section are issued by the Deployment Engine **de_lsapp** command.

ACUEX1505E Cannot create query.

Explanation: The **de_lsapp** command failed. An error occurred while trying to query the Deployment Engine installation database. No results were returned.

User response: See the message log for previous errors that caused this message.

ACUEX1507E Cannot retrieve the query results from the database.

Explanation: The **de_lsapp** command failed. An error occurred while querying the Deployment Engine installation database for the list of applications. No results were returned.

User response: See the message log for previous errors that caused this message.

ACUEX1509E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX1510E Deployment Engine could not process the de_lsapp command.

Explanation: Deployment Engine received an unexpected Java exception while processing the **de_lsapp** command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX1511E The -field option has more values than are allowed.

Explanation: The number of values that you specified with the **-field** option is greater than the number allowed by the **de_lsapp** command.

User response: To display syntax and option information for this command, issue the command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

de_lsfeat messages

The messages in this section are issued by the Deployment Engine **de_lsfeat** command.

ACUEX1305E Cannot create query.

Explanation: The **de_lsfeat** command failed. An error occurred while trying to query the Deployment Engine installation database. No results were returned.

User response: See the message log for previous errors that caused this message.

ACUEX1307E Cannot retrieve the query results from the database.

Explanation: The **de_lsfeat** command failed. An error occurred while querying the Deployment Engine installation database for the list of features. No results were returned.

User response: See the message log for previous errors that caused this message.

ACUEX1309E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX1310E Deployment Engine could not process the de_lsfeat command.

Explanation: Deployment Engine received an unexpected Java exception while processing the **de_lsfeat** command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

de_ismaint messages

The messages in this section are issued by the Deployment Engine `de_ismaint` command.

ACUEX1355E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX1356E Deployment Engine could not process the `de_ismaint` command.

Explanation: Deployment Engine received an unexpected Java exception while processing the `de_ismaint` command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX1357E The `-field` option has more values than are allowed.

Explanation: The number of values that you specified with the `-field` option is greater than the number allowed by the `de_ismaint` command.

User response: To display syntax and option information for this command, issue the command with the `-help` option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1358E Cannot create query.

Explanation: The `de_ismaint` command failed. An error occurred while trying to query the Deployment Engine installation database. No results were returned.

User response: See the message log for previous errors that caused this message.

ACUEX1360E Cannot retrieve the query results from the database.

Explanation: The `de_ismaint` command failed. An error occurred while querying the Deployment Engine installation database for the installed maintenance. No results were returned.

User response: See the message log for previous errors that caused this message.

de_lsproc messages

The messages in this section are issued by the Deployment Engine **de_lsproc** command.

ACUEX1455E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX1456E Deployment Engine could not process the **de_lsproc** command.

Explanation: Deployment Engine received an unexpected Java exception while processing the **de_lsproc** command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

de_restoredb messages

The messages in this section are issued by the Deployment Engine `de_restoredb` command.

ACUEX1556E Deployment Engine could not process the `de_restoredb` command.

Explanation: Deployment Engine received an unexpected Java exception while processing the `de_restoredb` command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX1559E The name specified for the backup file, *file_name*, is a directory name.

Explanation: The backup file name that you specified in your `de_restoredb` command is the name of a directory on the computer. The name of the backup file cannot be the same as the name of an existing directory.

User response: Reissue your command with a unique backup file name that does not match the name of a directory on the computer.

ACUEX1560E The backup file *file_name* does not exist.

Explanation: The file name you specified in the `-bfile` option cannot be found on the computer. You might have specified the wrong file or typed the file name incorrectly.

User response: Reissue your command with the name of a backup file that you are sure exists on the computer.

ACUEX1562E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX1563E You do not have the required permission to perform this operation. Ask your system administrator if you can be granted the necessary authorization.

Explanation: The user mode of the Deployment Engine run-time environment that you are using determines the permissions you need to issue this command. For a multiuser mode Deployment Engine run-time environment, only a root user can issue this command. For a single-user mode Deployment Engine run-time environment, a nonroot user can issue this command.

User response: If you do not know whether your copy of Deployment Engine was installed in multiuser mode or single-user mode, you can find out by issuing the `de_version` command. The `de_version` command displays the Deployment Engine user mode as well as the version.

For more details about command authorization, refer to the chapter on working with commands in the *Autonomic Deployment Engine for Administrators* book.

After determining the permissions required, contact your system administrator to request any additional authorization needed to run this command.

ACUEX1564E There was an error starting the database service.

Explanation: The database cannot be started at this time.

User response: See the message log for more information about the error. You can try to start the database manually by issuing the `acsisrv.sh -start` command on UNIX-based operating systems or the `acsisrv.bat -start` command on Windows operating systems.

ACUEX1565E

ACUEX1565E There was an error stopping the database service.

Explanation: The database cannot be stopped at this time.

User response: See the message log for more information about the error.

de_setdb messages

The messages in this section are issued by the Deployment Engine `de_setdb` command.

ACUEX1608E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX1609E Deployment Engine could not process the `de_setdb` command.

Explanation: Deployment Engine received an unexpected Java exception while processing the `de_setdb` command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX1611W Request failed to complete.

Explanation: The `de_setdb` command failed.

User response: See the message log for more information about the error.

ACUEX1612W The specified port number is not valid.

Explanation: The port number specified in the command is not valid. The value of the `-port` option must be a number from 1024 to 65534.

User response: See the message log for more information about the error.

ACUEX1613W The password reset was not successful.

Explanation: The password for the Deployment Engine installation database was not changed.

User response: See the message log for more information about the error.

de_uninstapp messages

The messages in this section are issued by the Deployment Engine **de_uninstapp** command.

ACUEX1761E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the supporting personnel for your application and provide this information when requesting assistance.

ACUEX1762E Deployment Engine could not process the **de_uninstapp** command.

Explanation: Deployment Engine received an unexpected Java exception while processing the **de_uninstapp** command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX1765E The application is not uniquely specified. Use both the **-uuid** and **-disc** options.

Explanation: Deployment Engine found more than one application instance that matched the **-uuid** or **-disc** option that you specified.

User response: Specify both the **-uuid** and **-disc** options on the **de_uninstapp** command to identify a unique application instance.

ACUEX1766E Deployment Engine cannot find the specified application.

Explanation: No application instance was found that matched the **-uuid** option, **-disc** option, or both, as specified in the command.

User response: Try your command again with **-uuid** and **-disc** options that match an instance of a known, installed application. To obtain a list of application instances currently installed on the computer, issue the **de_lsapp** command without any options. Any application instances returned will also include their UUID and discriminant values.

ACUEX1767E Deployment Engine cannot uninstall the specified application.

Explanation: The **de_uninstapp** command failed.

User response: See the message log for previous errors that caused this message.

ACUEX1769E The application that you specified is aggregated by another application. Aggregated applications cannot be directly removed.

Explanation: The application that you specified in the **de_uninstapp** command was installed as part of a larger base application. The specified application can only be uninstalled by uninstalling the larger base, or aggregating, application.

User response: To uninstall your application, use the **de_uninstapp** command to uninstall the aggregating application. Both applications will be removed as one.

ACUEX1771E Deployment Engine is not responding. Verify that the ACSI service is running.

Explanation: While processing the **de_uninstapp** command, the Deployment Engine database service could not be reached.

User response: Ensure that the ACSI service is currently running on the computer. Then try your command again.

de_uninstfeat messages

The messages in this section are issued by the Deployment Engine **de_uninstfeat** command.

ACUEX1661E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX1662E Deployment Engine could not process the **de_uninstfeat** command.

Explanation: Deployment Engine received an unexpected Java exception while processing the **de_uninstfeat** command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX1663E Found unknown option *option_name*.

Explanation: The option name specified is not a valid option for the command.

User response: To display syntax and option information for this command, issue this command with the **-help** option only. For even more details about how to use this command, refer to the chapter on administration commands in the *Autonomic Deployment Engine for Administrators* book.

ACUEX1665E The application is not uniquely specified. Use both the **-uuid** and **-disc** options.

Explanation: Deployment Engine found more than one application instance that matched the **-uuid** or **-disc** option that you specified.

User response: Specify both the **-uuid** and **-disc** options on the **de_uninstfeat** command to identify a unique application instance.

ACUEX1666E Deployment Engine cannot find the specified application.

Explanation: No application instance was found that matched the **-uuid** option, **-disc** option, or both, as specified in the command.

User response: Try your command again with **-uuid** and **-disc** options that match an instance of a known, installed application. To obtain a list of application instances currently installed on the computer, issue the **de_lsapp** command without any options. Any application instances returned will also include their UUID and discriminant values.

ACUEX1667E Deployment Engine cannot uninstall the specified feature.

Explanation: The **de_uninstfeat** command failed.

User response: See the message log for previous errors that caused this message.

ACUEX1668E The application that you specified is aggregated by another application. Features in aggregated applications cannot be directly removed.

Explanation: The application that you specified in the **de_uninstfeat** command was installed as part of a larger base application, called the aggregating application. The feature that you want to remove is installed as part of a smaller, aggregated application. Any modification to this smaller, aggregated application must be handled through modifications to the top-level aggregating application. The feature that you want to remove can only be removed by uninstalling its corresponding feature from the top-level aggregating application.

ACUEX1669E • ACUEX1674E

User response: To remove your feature, use the **de_uninstfeat** command to uninstall its corresponding feature from the top-level aggregating application. Both features will be removed as one. **Attention:** This action may remove other features as well.

ACUEX1669E The features were not uninstalled. Some or all of the features that you specified are not valid for the operation you requested.

Explanation: The features that you wanted to uninstall could not be removed. This message is always followed by an additional error message.

User response: Refer to the user response that is provided for the subsequent error message. Follow the advice in the user response or remove the features that are not valid from the list of features that you wanted to uninstall.

ACUEX1670E The following features are not installed or do not exist:
feature_list

Explanation: One or more features that you specified for removal do not have valid feature names. The features listed in the message are not installed for the application.

User response: Reissue your command specifying only valid features.

ACUEX1671E To remove the features that you specified, the following dependent features must also be removed:
feature_list

If you reissue your command with the **-auto** option, Deployment Engine will automatically remove the dependent features, along with the specified features, for you. Otherwise, you must first use the **de_uninstfeat** command to remove the above-listed features, and then reissue your original command again.

Explanation: Other features depend on the features that you specified for removal. These dependent features will not function correctly if the specified features are removed.

User response: To remove the specified features together with the dependent features, reissue your original **de_uninstfeat** command along with the **-auto** option. Otherwise, use the **de_uninstfeat** command to remove the dependent features listed in the message, and then reissue your original command again.

ACUEX1672E The following features are required and cannot be removed:
feature_list

Explanation: The features listed in the message are needed for the proper operation of the application. Removing them independently from removing the application is not allowed.

User response: To remove the required features, you must remove the full application. This can be done with the **de_uninstapp** command.

ACUEX1673E The following features were specified with the **-featid** option multiple times:
feature_list

Explanation: When specifying a feature with the **-featid** option, you can only specify the feature once. The message lists any features that were specified more than once.

User response: Retry your command with the duplicate features removed.

ACUEX1674E The following features failed feature validation rules:
feature_list

Explanation: Features are installed and uninstalled based on their conformity to certain validation rules. The features listed in the message were not uninstalled, because doing so would break these rules.

User response: Retry your command with the listed features removed.

ACUEX1675E A specified feature failed validation rules.

When feature *featureID_a* is not installed, feature *featureID_b* must be installed.

Explanation: This feature is part of a relationship that requires either it or another feature to be installed. Having neither feature installed is not valid.

User response: Before removing the feature that you specified in your original command, install the related feature named in the message. Then try your command again.

ACUEX1676E Features were not removed because the application specified in the command has no features installed.

Explanation: The specified application does not have any installed features. There is nothing to remove.

User response: Make sure you specified the correct application in your command.

ACUEX1677E Uninstalling the specified features will leave the following features without content:
feature_list

Explanation: The features you are trying to remove are subfeatures of other features. Removing these subfeatures will leave the features listed in the message empty. Deployment Engine rules do not permit empty features. You can, however remove the subfeatures along with their parent features, which are listed in the message. That way, no empty features will remain.

User response: Retry your original command, but also include the features listed in the message.

de_uninstmaint messages

The messages in this section are issued by the Deployment Engine **de_uninstmaint** command.

ACUEX1705E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX1706E Deployment Engine could not process the **de_uninstmaint** command.

Explanation: Deployment Engine received an unexpected Java exception while processing the **de_uninstmaint** command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

ACUEX1708E Deployment Engine returned the following message:

de_msg

Explanation: An unexpected Deployment Engine message was received while processing the **de_uninstmaint** command.

User response: See the message log for more information about the error.

ACUEX1709E *de_msg*

Explanation: An unexpected Deployment Engine error was received while processing the **de_uninstmaint** command. This message shows you the text of the message associated with that Deployment Engine error.

User response: See the message log for more information about the error.

ACUEX1711W Request failed to complete.

Explanation: The **de_uninstmaint** command failed.

User response: See the message log for more information about the error.

de_version messages

The messages in this section are issued by the Deployment Engine **de_version** command.

ACUEX1805E A processing error occurred within Deployment Engine that you cannot resolve by yourself.

Explanation: An unexpected Java exception was received from a Deployment Engine API. Fixing the problem requires outside assistance.

User response: Check the message and trace logs for additional details about the exception. Then contact the support personnel for your application and provide this information when requesting assistance.

ACUEX1806E Deployment Engine could not process the **de_version** command.

Explanation: Deployment Engine received an unexpected Java exception while processing the **de_version** command. This message is logged with additional information about the exception. (Note: Depending on the logging level that is currently enabled, the trace log might provide even more details about this exception.)

User response: Check the message and trace logs for additional details about the exception. If you still cannot determine the cause of the problem from the exception details provided in the logs, contact the support personnel for your application to obtain assistance.

Chapter 12. Trace logging

The trace log file contains information that you can use to troubleshoot problems yourself, or that you can send to IBM Support so that IBM can troubleshoot the problem.

Deployment Engine uses the Java-defined standard logging levels to control trace logging output for the Deployment Engine commands. These logging levels are ordered, so that enabling trace logging at a particular logging level also enables trace logging at all the higher levels. In addition to the ordered logging levels (listed below), there is an ALL level that you can specify to enable *all* available levels of trace logging. ALL is the default trace level.

The trace logging levels, in descending order, are as follows:

SEVERE

(Highest-level tracing.) Enables tracing of error messages, which include messages about serious failures that affect normal program processing.

WARNING

Enables tracing of warning messages that identify potential problems that the administrator should know about.

INFO

Enables tracing of informational messages that the administrator should know about.

CONFIG

Enables tracing of certain configuration data that might help debug problems which could be related to particular configurations.

FINE

Enables tracing of the lowest-volume, but most important, trace output.

FINER

Enables tracing of somewhat more-detailed trace output.

FINEST

(Lowest level tracing.) Enables tracing of the highest-volume, most-detailed trace output.

To set the trace level, modify the following line in the \$ACU_COMMON/deexlogger.properties file to indicate the trace logging level that you want. In the following example, the trace logging level is set to the default, ALL:

```
com.ibm.ci.log.trace.handlerLevel=ALL
```

The trace log contains information that helps you trace the flow of execution in order to diagnose problems. Each trace entry includes the following fields:

Trace entry date

Indicates the year, month, and day that the trace entry was generated.

Trace entry time

Indicates the time of day that the trace entry was generated.

Java class name

Indicates the name of the Java class that generated the trace entry.

Method name

Indicates the name of the method that generated the trace entry.

Trace logging level

Indicates the Java-defined standard logging level for the trace entry.

Message identifier

Indicates the identifier of the message for the trace entry.

Message text

Indicates the text of the message for the trace entry.

Chapter 13. Troubleshooting

The following topics provide troubleshooting information to help prevent or resolve problems related to Deployment Engine.

- **No matches were found when querying the database.**

Check the options that you entered with your **de_lsapp**, **de_lsmaint**, or **de_lsfeat** command for correctness and try again. You can issue the **de_lsapp** command *without options* to list all deployed applications and enough supplemental information to determine if the values that you used in your original command were correct.

- **Permission to process a command was denied.**

By issuing the current command, you are attempting an operation that you are not authorized to perform. Read “Command authorization” on page 21 along with the “Authorization” section for the command you are trying to run. Then obtain the correct authority and issue your command again.

- **The installation database is locked because it is being used by another process.**

Certain commands lock the Deployment Engine installation database to prevent simultaneous use by other commands. Wait until all currently running commands are finished before attempting to run your command again. If you are sure no other commands are currently being processed, then it is possible the database lock files have not been removed properly after the previous command. To correct this problem, try the following steps:

1. Check the logs directory, which can be found as described in Chapter 8, “Locating the Deployment Engine log files,” on page 61, for files similar to `.lock_0_1`, `.lock_0_2`, `.lock_1_2`, and the like. If you find lock files like these, remove them all.

2. Check for the file:

```
user.home/IBM/acuex/workspace/.metadata/.plugins/  
com.ibm.ci.siex.cli.main/.siex_lock
```

If you find this lock file, remove it.

3. After removing all the found lock files, try issuing your command again.

- **Maintenance installation was unsuccessful.**

A maintenance installation failure is most often caused when the maintenance:

- Has been targeted for a specific version of the application, which does not match the currently installed version.
- Requires a specific version of another application to be installed before the maintenance can be applied.
- Is attempting to install an application that is part of, or aggregated by, another application.
- Requires that another, already installed application be removed before the maintenance can be applied.
- Requires more disk space than is currently available.

Check your application documentation, maintenance documentation, or both for current requirements. In addition, look for solutions to your problem by checking for messages in the message log and then reading the corresponding message explanations and user responses provided in Chapter 11, “Messages issued by commands,” on page 75. You can also check the documentation for

your command that is provided in this book. This documentation covers the correct command and command option usage.

- **Maintenance removal was unsuccessful.**

A maintenance uninstallation failure is most often caused by one or more of the following things:

- Additional maintenance was applied to the target application *after* the maintenance that you want to remove was applied. To function properly, that additional maintenance depends on the presence of the maintenance you are trying to remove.
- Another application requires a specific version or fix level of the current application. Removing the maintenance that you specified will violate that requirement.
- The specified maintenance belongs to an aggregated application—an application that was deployed as part of another, parent application.
- The maintenance has been marked by the maintenance authors as “not removable.”

Many of these issues could be resolved by using the **-force** or **-auto** options for your command, but these options should be used with caution. Look for other alternatives by checking for messages in the message log and then reading the corresponding message explanations and user responses provided in Chapter 11, “Messages issued by commands,” on page 75. You can also check the documentation for your command that is provided in this book. This documentation covers the correct command and command option usage, and provides additional guidance on **-force**, **-auto**, and other options

- **Feature removal was unsuccessful.**

One of the feature selection rules will be violated if the specified feature is removed. If additional features need to be removed as well, you can try issuing this command with the **-auto** option to remove them. Check the documentation for your command that is provided in this book. This documentation covers the correct command and command option usage, and provides additional guidance on the **-auto** option. If an additional feature needs to be added in order to remove the specified feature, you must use the original installation media and installer to perform that operation.

- **Cannot schedule a database backup or file scan.**

Your command can only be run against an instance of Deployment Engine that was installed in multiuser mode. See “Command authorization” on page 21 along with the “Authorization” section for the command for more information.

- **Cannot set the database properties.**

Your command can only be run against an instance of Deployment Engine that was installed in multiuser mode. See “Command authorization” on page 21 along with the “Authorization” section for the command for more information.

- **Application removal was unsuccessful.**

The application that you want to remove could be part of another, aggregating application or required by another application. Check your application documentation, maintenance documentation, or both for current requirements. In addition, look for solutions to your problem by checking for messages in the message log and then reading the corresponding message explanations and user responses provided in Chapter 11, “Messages issued by commands,” on page 75. You can also check the documentation for your command that is provided in this book. This documentation covers the correct command and command option usage.

Chapter 14. Limitations

The Deployment Engine software related to administration and administration commands has the following known limitations:

- Only one change request can be processed at a time.
- Maintenance cannot be directly applied to contained, or child, applications that are included as part of other application packages—such as with applications in a software solution or suite. Maintenance for the contained application must be applied as maintenance to the parent application.
- Currently, applying or removing maintenance in the form of a full update is not supported by the **de_instmaint** and **de_uninstmaint** commands. Nor do these commands support migration actions. Further, only one incremental update can be applied or removed at one time, and only one application can be updated or targeted at one time.
- A full update cannot be uninstalled with the **de_uninstmaint** command. When the full update was originally deployed, it either completely replaced an existing application or it performed a fresh installation of an application that did not previously exist. The only way to remove a full update is to uninstall the entire application—which, in fact, is the full update. To remove the complete contents of the full update, you must remove the application itself by using the **de_uninstapp** command.

Also, see “Access restrictions in multiuser mode” on page 15 and “Access restrictions in single-user mode” on page 15 for database access restrictions.

Part 4. Appendixes

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Open source license notices

This section contains details concerning certain notices IBM must provide to you under its license to certain software code. The relevant notices are provided or referenced below. Please note that any non-English version of the licenses below is unofficial and is provided to you for your convenience only. The English version of the licenses below, provided as part of the English version of this file, is the official version.

Notwithstanding the terms and conditions of any other agreement you may have with IBM or any of its related or affiliated entities (collectively "IBM"), the third party software code identified below are "Excluded Components" and are subject to the terms and conditions of the License Information document accompanying this Program.

Apache Software License, Version 1.1

Copyright © 2000 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)."
Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Portions of this software are based upon public domain software originally written at the National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign.

Apache Software License, Version 2.0

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - a. You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - b. You must cause any modified files to carry prominent notices stating that You changed the files; and
 - c. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - d. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

W3C Software Notice and License

<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C[®] Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION. The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

Trademarks

IBM, the IBM logo, Cloudscape, OS/400, Tivoli, and WebSphere are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft and Windows NT, and Windows 2000 are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

A

- action descriptors 7
- ACU_COMMON environment variable vi, 61
- application packages
 - definition 5, 12
- applications
 - definition 5
 - querying default information example 31
 - removing 38
 - removing an application example 39
 - removing features example 41
- authorities
 - multiuser mode 14
 - single-user mode 14

B

- backups
 - initiating a database backup example 47
 - installation database 46
 - listing 52
 - querying scheduled database backups example 52
 - restoring database backup example 55
- base applications
 - definition 11
- base software packages
 - definition 11
- bold typeface, meaning of vi
- bootstrap program 8

C

- case sensitivity in command syntax 23
- change manager
 - in Deployment Engine operating environment 8
 - messages 69
- change requests
 - definition 5
- command line invocation messages 88
- commands 23, 24
 - application-related commands 27
 - case sensitivity in 23
 - de_backupdb command 46
 - de_fs command 49
 - de_help command 24
 - de_instmaint command 28
 - de_lsapp command 30
 - de_lsfeat command 33
 - de_lsmaint command 36
 - de_lsproc command 52
 - de_restoredb command 54
 - de_setdb command 56
 - de_uninstapp command 38
 - de_uninstfeat command 40
 - de_uninstmaint command 42
 - de_version command 58
 - Deployment Engine–related commands 45
 - location of command files 22
 - special characters in command syntax statements 22
 - summary of administration commands 19

- commands (*continued*)
 - ways to display help 24
- common messages
 - for Deployment Engine commands 76
 - for Deployment Engine components 68
- components
 - of Deployment Engine operating environment 6
- configuration unit deployment descriptors 7
- configuring
 - definition 6
- conventions used in this book
 - command syntax 22
 - typeface vi
 - UNIX directory names vi
 - UNIX directory paths vi
 - UNIX environment variables vi

D

- de_backupdb command
 - messages 95
 - syntax and usage 46
- de_fs command
 - messages 97
 - syntax and usage 49
- de_help command
 - command syntax and usage 24
 - messages 99
- de_instmaint command
 - messages 100
 - syntax and usage 28
- de_lsapp command
 - messages 101
 - syntax and usage 30
- de_lsfeat command
 - messages 102
 - syntax and usage 33
- de_lsmaint command
 - messages 103
 - syntax and usage 36
- de_lsproc command
 - messages 104
 - syntax and usage 52
- de_restoredb command
 - messages 105
 - syntax and usage 54
- de_setdb command
 - messages 107
 - syntax and usage 56
- de_uninstapp command
 - messages 108
 - syntax and usage 38
- de_uninstfeat command
 - messages 109
 - syntax and usage 40
- de_uninstmaint command
 - messages 112
 - syntax and usage 42
- de_version command
 - messages 113
 - syntax and usage 58

- dependency checker
 - in Deployment Engine operating environment 8
 - messages 70
- dependency checking 28, 42
- deploying
 - definition 6
- Deployment Engine
 - definition 3
 - messages from commands 75
 - messages from components 67
 - operating environment components 6
 - previous name (Solution Install) 3
 - run-time components 7
 - what it does 4
- Deployment Engine APIs
 - in Deployment Engine operating environment 8
- Deployment Engine run-time environment
 - definition 13
 - in the application package 13
- Deployment Engine-enabled application
 - definition v
- descriptors
 - in Deployment Engine operating environment 7

F

- features
 - definition 3
 - displaying information about 33
 - querying feature information example 34
 - removing 40
- file scans
 - definition 49
 - initiating a file scan example 51
 - querying scheduled file scans example 52
 - scheduling 49
- fix software packages
 - definition 12
- fresh installation
 - definition 11
- full update software packages
 - definition 11

G

- guidance information 3

H

- help information, displaying 24
- hosted resources
 - in Deployment Engine operating environment 9
- hosting environments
 - in Deployment Engine operating environment 9

I

- incremental software packages
 - definition 12
- installable unit deployment descriptors 7
- installable units
 - definition 4
- installation database
 - access restrictions in multiuser mode 15
 - access restrictions in single-user mode 15

- installation database (*continued*)
 - and file scans 49
 - backing up 46
 - changing port number 56
 - changing port number example 57
 - database access 15
 - in Deployment Engine operating environment 8
 - initiating a database backup example 47
 - messages 71
 - multiuser access 14
 - querying scheduled database backups example 52
 - resetting password 56
 - restoring database backup example 55
 - restoring from copy 54
- instance
 - definition 5
- integrity checking 28, 42
- introduction to Deployment Engine 3
- italic typeface, meaning of vi

L

- legal notices
 - IBM 123
 - third party 124
- levels, trace 115
- limitations of the Deployment Engine administration
 - commands 119
- logging
 - trace logging 115

M

- main messages 88
- maintenance
 - definition 3
 - deploying 28
 - displaying information about 36
 - installing maintenance example 29
 - querying maintenance information example 37
 - removing 42
 - removing maintenance example 44
- media descriptors 7
- message log fields 64
- messages
 - change manager component 69
 - command line invocation (main messages) 88
 - common messages
 - for Deployment Engine commands 76
 - for Deployment Engine components 68
 - de_backupdb messages 95
 - de_fs messages 97
 - de_help messages 99
 - de_instmaint messages 100
 - de_lsapp messages 101
 - de_lsfeat messages 102
 - de_lsmaint messages 103
 - de_lsproc messages 104
 - de_restoredb messages 105
 - de_setdb messages 107
 - de_uninstapp messages 108
 - de_uninstfeat messages 109
 - de_uninstmaint messages 112
 - de_version messages 113
 - dependency checker component 70

messages (*continued*)
format
 message help 64
 message numbers 63
help information for 67, 75
installation database component 71
message log fields 64
message text 64
operating system touchpoint component 73
monospace font, meaning of vi
multiuser mode 14

N

nonroot user
 and Deployment Engine in single-user mode 14
 definition 21
notices
 IBM 123
 third party 124

O

operating environment of Deployment Engine 6
operating system touchpoint
 in Deployment Engine operating environment 9
 messages 73
overview of Deployment Engine 3

P

packaging
 definition 5
payload files
 definition 3
 in Deployment Engine operating environment 8
problem determination 61
 Deployment Engine command messages 75
 Deployment Engine component messages 67
 finding the logger.properties files
 component messages 61
 message help 64

R

representing strings that include spaces 24
root user
 and Deployment Engine in multiuser mode 14
 definition 21
rules for combining command options 23

S

SI_PATH environment variable vi
single-user mode 14
software
 definition 5
software deployment
 definition 3
software deployment programs
 and bootstrap program 8
 in Deployment Engine operating environment 8
software package types
 base 11
 definition 11

software package types (*continued*)
 fix 12
 full update 11
 incremental update 12
software packages
 base software package 11
 definition 3, 11
 fix software package 12
 fresh installation 11
 full update software package 11
 in Deployment Engine operating environment 7
 in the application package 11
 incremental update software package 12
 types 11
specifying an application uniquely 23
syntax statements, command 22

T

terminology, introductory 3, 5
touchpoints
 in Deployment Engine operating environment 9
trace levels 115
trace loggers
 trace log fields 115
trace logging
 description 115
 setting the trace level 115
troubleshooting
 tips 117
typeface conventions vi

U

UNIX conventions used in this book vi
user modes
 displaying 58
 multiuser mode 14
 querying Deployment Engine user mode example 58
 single-user mode 14

V

variables
 ACU_COMMON environment variable vi, 61
 SI_PATH environment variable vi
versions
 displaying Deployment Engine version 58
 querying Deployment Engine version example 58

W

Web service interface
 in Deployment Engine operating environment 9
WebSphere touchpoint
 in Deployment Engine operating environment 9



Printed in USA