

XDE Project Management Services

System Administrator's Guide

XEROX

610E00220
September 1985

**Xerox Corporation
Office Systems Division
2100 Geng Road
MS 5827
Palo Alto, California 94303**

Copyright © 1985, Xerox Corporation. All rights reserved.
XEROX®, 8010, and XDE are trademarks of XEROX CORPORATION.

Printed in U.S. A.



Table of contents

1	Introduction	
1.1	Adobe system administration	1-1
1.2	Librarian system administration	1-2
2	Adobe service	
2.1	Terminology	2-1
2.2	Adobe service commands	2-2
2.3	Setting up an Adobe database	2-4
2.3.1	Defining a database	2-5
2.3.2	File service space for the database records	2-6
2.3.3	The System Description file	2-6
2.3.4	Example	2-9
2.3.5	Putting the description on the Adobe service	2-11
2.3.6	Registering the Adobe service with the Clearinghouse	2-12
2.3.7	Setting other database parameters	2-12
2.4	Maintaining an Adobe database	2-12
2.4.1	Viewing the status of an Adobe service	2-12
2.4.2	Backup and Accelerator file integrity	2-13
2.4.3	Viewing the parameters and status of an Adobe database	2-15
2.4.4	Changing the parameters of an Adobe database	2-16
2.4.5	Moving an Adobe database	2-17
2.4.6	Destroying an Adobe database	2-17
2.4.7	Changing a database's fieldlist	2-18
3	Librarian service	
3.1	Introduction	3-1
3.2	Setting up your Librarian service	3-1

Table of contents

3.2.1	Installing a Librarian service	3-1
3.2.2	Creating a new database	3-2
3.2.3	Database parameters	3-2
3.2.4	Listing and tracing.	3-3
3.3	Maintaining your Librarian service	3-3
3.3.1	Backup and recovery	3-3
3.3.2	Deleting and moving databases	3-4
3.4	Librarian service command index.	3-4



Introduction

Adobe and the Librarian, the XDE project management tools, operate as services available to users on the local network. Some effort is required to keep these services functioning optimally, so that they best serve the needs of the user community.

The tasks of Adobe and Librarian system administrators are similar. Each must define the size and scope of the databases, considering the best configurations for their users and the task at hand. Understanding one of these systems can help in understanding the other, since both are based on the concepts of unique objects in databases, objects that represent files or activities related to system development.

1.1 Adobe system administration

The Adobe system administrator creates new Adobe databases. By defining certain characteristics, such as user interfaces of tools or screen layouts, the administrator helps database users get their tools into a form most helpful and understandable to them. Several Adobe databases can be created, thus dividing up a project system into manageable pieces. For example, one such division might be between hardware development issues and software development issues.

The Adobe system administrator sets up the databases but does not necessarily make any entries into them. No initial entries are required for an Adobe database to function properly after its characteristics have been defined in the descriptor file.

1.2 Librarian system administration

A Librarian system administrator sets up the Librarian database for the system (or for components of the system, if it is very large). But for the Librarian to work, each system object must have a libject to represent it in the database. The administrator is often the one who creates these libjects. Although implementors could also easily create libjects, it is sometimes more convenient to centralize their creation.



Adobe service

The Adobe service provides access to information about Adobe databases. The service maintains a system description and accelerator files associated with each Adobe database it controls. A system description includes accelerator files for the database's records along with logistic information about the database, such as the next request number and the directory where the records are stored.

2.1 Terminology

Adobe system - Another term for an Adobe database. It usually refers to the database's representation on the Adobe service.

AR - an Adobe Record or Action Request. An AR is a record in an Adobe database. It is represented as a file whose name is `nnnnn.ar` where `nnnnn` is the number assigned to the AR (with leading 0's prepended to make it a 5-digit number). This file is stored on the host and directory specified as part of the system description.

Accelerator file - An accelerator file exists for each field in a database's fieldlist except unbounded strings. It contains the values of all ARs for the field in question. These files speed access to the contents of the database's ARs.

Attribute - a value associated with a field in the fieldlist. Examples of field attributes include the name of the field and its type.

Field - one element of information contained in a database record.

Fieldlist - a list of the names and types of all the fields in a database. It includes other attributes of certain types of fields. For example, the maximum length of a bounded string field is a part of that field's information in the fieldlist. A fieldlist is part of the system description.

System parameter - a value associated with the system description of an Adobe database, such as the location of an AR or the next submit number for that database.

System Description - a set of fields describing the database. It includes the name of the database, its version, the file service and directory on which its records are stored, the fieldlist, and default information.

SystemMust default - a default value specified in the System Description file and subsequently filled in by the user with the value specified in the description for the tool type in question. A system default cannot be changed by the tool user.

Updater process - a background process associated with an Adobe system. It both performs backup for the system and assures integrity of the system's accelerator files.

2.2 Adobe service commands

This section explains each Adobe service command, listed below in alphabetical order. To use the Adobe service commands, you must be in the Adobe Service context. These commands will be discussed further in the sections on creating and maintaining Adobe databases.

Backup Systems

Causes all the files that the service maintains about a specified database to be backed up to a specified backup location when service is started. Available to the enabled user at all times.

Change FieldList

Allows changes to be made to the fieldlist of an Adobe database. Available to the enabled user when the service is started.

Change Location of ARs

Sets the location of ARs for the specified database to the host and directory given, causing the version number of the system description to be incremented. Available to the enabled user when the Adobe service is started.

Destroy AR System

Deletes all the database's files on the server after asking whether the user would like any of the database's files backed up. This command is used when moving a database to a different Adobe service or when permanently destroying a database. This command has no effect on the database's AR files. Available to the enabled user when the Adobe service is stopped.

Examine Update List

Displays the current contents of the updater process's list of ARs to be updated for the specified database. Available when the Adobe service is started.

Find Database/Service

Returns the name of the Adobe service on which the Adobe database you name resides, or returns the names of the Adobe databases residing on the Adobe service

you name. Available to the logged-on user when the Adobe service is started or stopped.

Get Default User File

Retrieves the default user file for a particular Adobe database. Available to the enabled user when the Adobe service is started.

List Systems

Displays the Adobe systems (databases) that this Adobe service supports. Available when the Adobe service is started.

Make Accelerator File

Causes the contents of one field's accelerator file to be assumed invalid and completely recreated. This command typically is used when an accelerator file is believed to be damaged in some way. Available to the enabled user when the Adobe service is started.

Put Default User File

Sets the default user file for an Adobe database. Available to the enabled user when the Adobe service is stopped.

Put Submit Number

Allows the user to specify the number assigned to the next AR submitted. Available to the enabled user when the Adobe service is started.

Put System Description

Establishes an Adobe database described by a description file on the Adobe service. Available to the enabled user when the Adobe service is started.

Register System

Available to the enabled user when the Adobe service is started. Registers a database with the Clearinghouse, associating this Adobe service with the database given.

Restore ARSystem

Retrieves all the files needed to recreate an Adobe database from the specified source location. This command is useful for moving an Adobe database to a new Adobe service or restoring a database if it has been damaged. Available to the enabled user when the Adobe service is started.

Set Start Time

Sets the starting time for periodic update of the specified database's accelerator files. Available to the enabled user when the Adobe service is started.

Set Stop Time

Sets the stopping time for periodic update of the specified database's accelerator files. Available to the enabled user when the Adobe service is started.

Show Accelerator Activity

Displays the current accesses being made to the accelerators of the specified database. Available when the Adobe service is started.

Show Client Activity

Displays the number of clients currently accessing the Adobe service. Available when the Adobe service is started.

Show System Parameters

Displays the values of the specified database's operating parameters. Operating parameters include the next submit number, the starting and stopping time for the periodic update, and the location of this database's ARs. Available when the Adobe service is started.

UnRegister System

Deletes a database's entry from the Clearinghouse. Available to the enabled user when the Adobe service is started.

Update All ARs

Indicates to the updater process that all the accelerator entries for all ARs should be considered out of date and therefore updated. Available to the enabled user when the Adobe service is started.

Update Specified ARs

Indicates to the updater process that all the ARs in the range given should be considered out of date and therefore updated. Available to the enabled user when the Adobe service is started.

2.3 Setting up an Adobe database

There are six steps to setting up a new Adobe database:

1. Define the database contents precisely.
2. Procure FS space for the database records.
3. Create a system description file.
4. Put the database description on an available Adobe service.
5. Register the database with the Clearinghouse, using the Register command on the Adobe service.
6. Set any system parameters not in the system description

2.3.1 Defining a database

Spending some thought defining an Adobe database will save you time in the long run. Although the Adobe service provides some facilities for modifying the definition of a database after it has been put into use (see 2.4.7), it is time consuming to make extensive changes. Work out what is wanted in the database. Observe what can and cannot be accommodated by the existing Adobe implementation. That way, you can create the best Adobe database to solve the problem. (In the past, poor design has been the most common cause for abandoning use of a database.)

Adobe supports six different kinds of database fields. They are described below.

- ARId:** There is one and only one **ARId** per Adobe system. The item of this type contains the unique number which identifies an AR in the database.
- Numeric:** This type is for fields that contain numeric information. Items of this type will not allow any non-numeric data to be entered into their fields.
- DateTime:** Fields of this type represent a time of day. Information entered into these fields should be of the form *dd-mmm-yy* followed optionally by *hh:mm:ss*. Dates entered with no time following will have 00:00:00 added to them. **DateTime** fields may be left empty, but an invalid **DateTime** will be replaced by the current date and time.
- BoundedString:** Bounded string fields have a specified length limit. By limiting string length, Adobe can maintain the data in these fields in accelerator files. Since these fields have accelerator files, they can be used as query keys. (See the section on Adobe Query in the Adobe Tools chapter of the *Project Management Tools User Manual*.)
- UnboundedString:** Unbounded string fields have no length limit. Since they are unlimited, they do not have accelerator files. They cannot be queried on.
- Enumerated:** Enumerated fields are of two types: independent and dependent. An independent item has a fixed set of values associated with it. A dependent item has several possible sets of values. Specifically, the value set for a dependent item changes to reflect the current value of its associated independent enumerated item. For example, if you change an independent enumerated item, any enumerated item dependent on it will be reset to **NIL**. The menu describing the dependent item's possible values will be changed accordingly. **NIL** is a special symbol indicating that there is no specified value. It is automatically provided as a possible value in all sets of enumerated values.

To define your database, make a list of the information you want to keep track of. (Refer to the Adobe Tools chapter in the *Project Management Tools User Manual* for an example of a

simple database description.) Once you decide on the information you want, select a name to refer to each piece of information. These names will be the field names for the fields in the database. Select the type of field best suited to represent each item of information. If an item is a bounded string, determine a suitable maximum length. For enumerated items, list the set of values the item has.

When formulating your description, create a brief user's reference. It need be no more detailed than the description of Personnel Records used as an example in the Adobe tools chapter of the *Project Management Tools User Manual*. This user's reference should contain the name and type of each item found in your database along with a brief description of the information contained in each field. A simple reference like this can be very valuable. Your users will understand better what should be entered in the fields, so there will be less chance of creating improper entries. It is also useful for you or for the future administrator of the database as a quick reference to the database's contents. It will help you see what could be added, removed, or changed in the system description to better serve the user community's continuing needs.

2.3.2 File service space for the database records

The records that compose an Adobe database are stored as files on a file service. One of the parameters of any database is the location of its records. This location is initially specified in the system description file (see the next section). You must create a file drawer or otherwise obtain space on a file service for your database's records. You should determine how much space to obtain, based on how many records you anticipate for your database, how large you expect these files to be, and how rapidly you expect the database to grow. If you locate your records on a file service that is already almost full, there may not be enough room when your database expands. If this happens, you may need to move your database (see section 2.4.5). As a guideline, a record in a medium-sized database (18-25 fields with two or three unbounded strings of only a few hundred characters) occupies two or three pages.

Adobe observes the access list controls maintained by a file service. Make sure that the access lists for the file drawer or folder are set appropriately. A user must have Add/Write access to submit and edit ARs and read access to examine ARs. Delete access should be given only to the owner of the file drawer, who is usually the system administrator for the database in question.

2.3.3 The System Description file

This section describes the syntax and semantics of files describing Adobe databases. Such files are text files that are created from scratch.

The description of an Adobe database consists of a number of sections, syntactically similar to sections in a *User.cm* file. The first section, entitled "SystemParams" is concerned with version information about the database. Each of the other sections describes one field of the database.

The first section contains the following entries:

```
[SystemParams]
SystemName: "MySystem:Domain:Organization"
```

Host: "FS:Domain:Organization"
Directory: "MySystemDirectory/Subdirectory"
Version: 1
NumberOfFields: 22

The **SystemName** is the fully qualified name of the database. The **Host** entry is the fully qualified name of the file server on which the records of the database are stored. The **Directory** entry is the name of the directory on the host where the database records are to be stored. This directory must exist on the file service. The **Version** entry is a cardinal number that represents the version of the Adobe system described by this file. **Version** should be 1 for a new database. The **NumberOfFields** entry indicates how many field declarations follow. (Sections that describe the same field, that is, dependent enumerated fields, are not counted individually. Instead, they are thought of as one field description.)

For each field description, there are required attributes, optional attributes, and type-dependent attributes. Field descriptions are separated by blank lines, so there can be no blank lines within a field description. Each description starts with the field name in square brackets

[Field Name]

Following the field name is a sequence of lines that describe the attributes of the field. These attributes may be defined in any order. The only restriction is that lines describing the same attribute must be grouped together. For example, all lines specifying Default must be together. Each line contains the name of the attribute, followed by a colon, then the attribute value

Attribute: value

The **Type** attribute must be specified for all fields. **Type** must have one of the following values: **ARId**, **DateTime**, **Numeric**, **Enumerated**, **BoundedString**, or **UnboundedString**.

If the type of the field is **BoundedString**, the field description must contain a **MaxLength** attribute. This attribute is a cardinal value indicating the maximum length of the bounded-length string.

If the type of the field is **Enumerated**, the field description must contain a set of **EnumeratedItem** attributes. These attributes describe the set of values that the enumerated field can take. (The value **NIL** is always a legal value for any enumerated field. It is implicit in the description. **NIL** should not be included in your description file.) There may be an arbitrary number of **EnumeratedItem** attributes per enumerated field, but they must appear consecutively in the field description. The value of the attribute is a record. The record must contain a **Name** field, which is a string indicating the name of the enumerated value. This name string must be contained in quotes if it has a space. The record also must contain a **key**, which is a cardinal value. The key value **0** is reserved for the **NIL** enumerated value. No two key values may be the same.

EnumeratedItem: [name: "Dog", key: 3]

Enumerated field descriptions must also contain a **DependsOn** attribute. The value of a **DependsOn** attribute is a record containing the name of the field and the item in the field

that this particular section depends on. If the field does not depend upon another enumerated field, these value are **NIL**.

DependsOn: [field: **NIL**, item: **NIL**] --no dependency

DependsOn: [field: **Section**, item: "System Software"] --depends on
EnumeratedItem *System Software* in the **Section** field

An enumerated field description that depends on another almost always needs to describe more than one set of **EnumeratedItem** attributes. That is, it should describe one set of attributes per **EnumeratedItem** value of the independent enumerated field. To create such a description, declare the field multiply with different **DependsOn** attributes. Refer to the example below:

[**Section**]

Type: **Enumerated**

DependsOn: [field: **NIL**, item: **NIL**]

EnumeratedItem: [name: "Services Software", key: 1]

EnumeratedItem: [name: "System Software", key: 2]

[**Area**]

Type: **Enumerated**

DependsOn: [field: **Section**, item: "Services Software"]

EnumeratedItem: [name: "Communication Services", key: 21]

EnumeratedItem: [name: "Basic Services", key: 22]

EnumeratedItem: [name: "Distributed Services", key: 23]

EnumeratedItem: [name: Other, key: 24]

[**Area**]

Type: **Enumerated**

DependsOn: [field: **Section**, item: "System Software"]

EnumeratedItem: [name: "IO Architecture", key: 31]

EnumeratedItem: [name: Mesa, key: 32]

EnumeratedItem: [name: Pilot, key: 33]

EnumeratedItem: [name: Other, key: 34]

A field description may also contain a set of default attributes for specifying **SystemMustDefault** values for the field in any of the released Adobe tools. The value of these default attributes is a record containing the tool name that should in turn represent the default and the default value.

Default: [tool: **Submit**, default:**CurrentTOD**]

The value of the default may be any legal value of the corresponding type. In addition, there are two distinguished values that the system recognizes. The string **CurrentTOD** is the current time of day, and the string **LoginName** is the name of the logged-in user and their domain and organization (for instance, Smith:OSBU North:Xerox).

2.3.4 Example

The following AR System Description describes Personnel Records, one of the Adobe database used as an example in the Adobe Tools chapter of the *Project Management Tools User Manual*.

[SystemParams]

SystemName: "Personnel Records:Domain:Organization"

Host: "FS:Domain:Organization"

Directory: "PersonnelRecords/ARs"

Version: 1

NumberOfFields: 14

[Number]

Type: ARId

[Last Name]

Type: BoundedString

MaxLength: 30

[First Name]

Type: BoundedString

MaxLength: 15

[Middle Initial]

Type: BoundedString

MaxLength: 1

[SSN]

Type: Numeric

[Hire Date]

Type: DateTime

[Section]

Type: Enumerated

DependsOn: [field: NIL, item: NIL]

EnumeratedItem: [name: "Advanced Development", key: 1]

EnumeratedItem: [name: "Services Software", key: 2]

EnumeratedItem: [name: "System Software", key: 3]

EnumeratedItem: [name: "Workstation Software", key: 4]

EnumeratedItem: [name: "Other", key: 5]

[Area]

Type: Enumerated

DependsOn: [field: Section, item: "Advanced Development"]

EnumeratedItem: [name: "Graphics & Printing", key: 11]

EnumeratedItem: [name: "International", key: 12]

EnumeratedItem: [name: "Networks", key: 13]

EnumeratedItem: [name: "User Interfaces", key: 14]

EnumeratedItem: [name: "Workstations", key: 15]

EnumeratedItem: [name: "Other", key: 16]

[Area]

Type: Enumerated
DependsOn: [field: Section, item: "Services Software"]
EnumeratedItem: [name: "Communication Services", key: 21]
EnumeratedItem: [name: "Basic Services", key: 22]
EnumeratedItem: [name: "Distributed Services", key: 23]
EnumeratedItem: [name: Other, key: 24]

[Area]

Type: Enumerated
DependsOn: [field: Section, item: "System Software"]
EnumeratedItem: [name: "IO Architecture", key: 31]
EnumeratedItem: [name: Mesa, key: 32]
EnumeratedItem: [name: Pilot, key: 33]
EnumeratedItem: [name: Other, key: 34]

[Area]

Type: Enumerated
DependsOn: [field: Section, item: "Workstation Software"]
EnumeratedItem: [name: "Basic Workstation", key: 41]
EnumeratedItem: [name: "Workstation Documents", key: 42]
EnumeratedItem: [name: "Workstation Functions", key: 43]
EnumeratedItem: [name: "Workstation Services", key: 44]
EnumeratedItem: [name: Other, key: 45]

[Area]

Type: Enumerated
DependsOn: [field: Section, item: Other]
EnumeratedItem: [name: Other, key: 51]

[Current Status]

Type: Enumerated
DependsOn: [field: NIL, item: NIL]
EnumeratedItem: [name: Active, key: 61]
EnumeratedItem: [name: Retired, key: 62]
EnumeratedItem: [name: Terminated, key: 63]
EnumeratedItem: [name: Resigned, key: 64]
EnumeratedItem: [name: Other, key: 65]

[Job Title]

Type: BoundedString
MaxLength: 40

[Grade]

Type: Numeric

[Other Notes]

Type: UnboundedString

[Edited-By]

Type: BoundedString
MaxLength: 30
Default: [tool: submit, default: LoginName]
Default: [tool: edit, default: LoginName]

[Edit-Date]
Type: DateTime
Default: [tool: submit, default: CurrentTOD]
Default: [tool: edit, default: CurrentTOD]

2.3.5 Putting the description on the Adobe service

When the system description file has been completed, you are ready to place it on an Adobe service. You may select an existing Adobe service to support your database, or you may bring up a new service, assuming you have a machine to function as a server. If you are bringing up a new Adobe service, you will need **AdobeService.bcd** and **AdobeSDF.bcd** from the release directory. Once you have determined the Adobe service with which your database will be associated, you are ready to proceed.

The **Put System Description** command is used to place an Adobe database's system description onto an Adobe Service. **Put System Description** takes your description file as input. This description file may be retrieved to the server or you can specify it remotely.

When the command asks **Parse Only?**, it wishes to know whether it should merely parse the description file given or whether should it proceed with putting the system description onto the Adobe Service (provided that the parse of the file succeeds). Answer **Yes** in response to the **Parse Only?** question until you are sure your file is formed properly. Doing so saves the **Put System Description** command some time.

In the Example 1 below, a user retrieves a description file from a remote file server, parses the description file to verify its validity, then puts the system description onto the Adobe service.

```
Adobe!Set Remote DirectoryReturn
  Remote Directory: (FS1:Domain:Organization)My Drawer/
    DescriptionsReturn
Adobe!Retrieve FilesReturn
  File List: PersonnelRecords.descReturn
PersonnelRecords.desc ... retrieved
Adobe!Put System DescriptionReturn
  Description File: PersonnelRecords.descReturn
  ParseOnly?: YReturn
Parsing Description...OK.
Adobe!Put System DescriptionReturn
  Description File: PersonnelRecords.descReturn
  ParseOnly?: NReturn
Parsing Description...ok.      Storing Description...ok.
Done.
Adobe!
```

Example 1

Errors may occur during the process of parsing your system description or putting it onto the Adobe service. Error messages will indicate the cause of the problem.

2.3.6 Registering the Adobe database with the Clearinghouse

You must register your Adobe database with the Clearinghouse to make it accessible to users. To register your Adobe database with the Clearinghouse, use the command **Register System**. Example 2 shows the registration of a system.

```
Adobe!Register SystemReturn
  Enter Fully-qualified DataBase Name: Personnel Records:OSBU North:
    XeroxReturn
  Registering Personnel Records:OSBU North:Xerox
  Confirm (Y/N): y
Done.
Adobe!
```

Example 2

If the command is unsuccessful, you will receive a message indicating the reason.

2.3.7 Setting other database parameters

You should make sure that you set start and stop times for periodic update of your database. If you fail to set these parameters, no periodic updating will be done on your database. See section 2.4.2 for details on **Set Start Time** and **Set Stop Time**.

Set the default user file for an Adobe system by using the **Put Default User File** command. Create a user file for the database, retrieve it to the service, and **Put Default User File**.

2.4 Maintaining an Adobe database

An Adobe database that is stable with respect to all its descriptions and parameters requires little maintenance; however, no database is likely to remain static. This section outlines the procedures for examining the status of databases and of the Adobe service on which they reside. It also tells how to make changes to existing databases.

2.4.1 Viewing the Status of an Adobe Service

A list of the databases residing on an Adobe service may be obtained by using the **List Systems** command at that service. Example 3 shows the use of the **List Systems** command.

```
Adobe!List SystemsReturn
  Database Name 1
  Database Name 2
  Database Name 3
Adobe>
```

Example 3

To obtain the list of databases on some other Adobe service or the name of the Adobe service on which a database resides, use the **Find Database/Service** command, illustrated

in Example 4. You can then see which Adobe service serves the database called Personnel Records. Then, you may ask what other databases are served by that service.

```
Adobe!Find DataBase/ServiceReturn
  Enter Fully-qualified DataBase or Service Name: Personnel Records:
    Domain:OrganizationReturn
  DataBase Service:Domain:Organization
  Adobe!Find Database/ServiceReturn
    Enter Fully-qualified DataBase or Service Name: DataBase Service:Domain:
      OrganizationReturn
  Personnel Records:Domain:Organization
  DataBase:Domain:Organization
  Adobe!
```

Example 4

To inspect the level of activity an Adobe service is experiencing at any particular time, use the command **Show Client Activity**. This command displays the number of remote clients for whom the service is processing requests. It is shown in Example 5.

```
Adobe!Show Client ActivityReturn
  Client Count: 0
  Adobe!
```

Example 5

2.4.2 Backup and Accelerator File Integrity

The Adobe service provides facilities for backing up information in Adobe systems and for ensuring the integrity of accelerator files.

An *updater process* is associated with each Adobe database existing on the Adobe service. This process performs several functions. First, it assures that anytime a change is made to an AR, the accelerator files are updated accordingly. Accelerator files are not automatically backed up, since they can be regenerated from the database's records. The updater process also performs ongoing integrity checks of the accelerator files.

The updater process is notified anytime a change is *about to be made* to an AR. If five minutes go by and the process is not notified that an AR was processed successfully, it assumes a change was made to the AR and the accelerator files were not properly updated. The process then updates the accelerator file entries for the AR in question.

This process also re-updates the accelerator files for all ARs during a span of time that the administrator may specify. The commands **Set Start Time** and **Set Stop Time** are used to indicate this updating period. The times given as arguments to these commands must be in the *hh:mm:ss* or *hh:mm* format where *hh* is specified using the 24-hour clock. For example, if you wish the Adobe service to perform this updating for your database between 11 p.m. and 4 a.m. each day, your commands to the service would be as shown in Example 6.

```

Adobe!Set Start Time Return
Adobe systems
  1. DataBase Name 1
  2. DataBase Name 2
Enter System number (1 2): 1Return
Enter Start Time: 23:00RETURN
Adobe!Set Stop Time
Adobe systems
  1. DataBase Name 1
  2. DataBase Name 2
Enter System number (1 2): 1Return
Enter Stop Time: 4:00RETURN
Adobe!

```

Example 6

You may sometimes wish to back up all the files that the service maintains regarding a database. This includes the system description file and default .user file, as well as all the accelerator files and a backing file maintained by the updater process. The **Backup Systems** command, illustrated in Example 7, provides this capability.

```

Adobe!Backup SystemsReturn
  1. DataBase Name 1
  2. DataBase Name 2
Enter System number: (1 2): 1Return
Backup path: (FS:Domain:Organization)DataBaseName1/Backup Return
Backup path is set to: (FS:Domain:Organization)DataBaseName1/Backup
Confirm (Y/N): y Return
Storing System Description ... Done
Storing .user file ... Done
Storing ... Updater Backing ... Done
Storing ... Accelerator Files ... Done
Adobe!

```

Example 7

2.4.3 Viewing the Parameters and Status of an Adobe Database

Each Adobe database has a number of system parameters associated with it. Some of them are determined from the system description file, some must be set explicitly, and others are set automatically. Most of them can be altered by a system administrator (see section 2.4.4). To view these system parameters, use the **Show System Parameters** command, illustrated in Example 8.

```
Adobe!Show System ParametersReturn
Adobe systems
 1. DataBase Name 1
 2. DataBase Name 2
 3. DataBase Name 3
 Enter System number (1 3): 1Return
System Parameters for DataBase Name 1
  Host: FS1:Domain:Organization
  Directory: DataBaseName 1/ARs
  Version: 4
  Next Submit Number: 1500
  Start Time: 22:00:00
  Stop Time: 03:00:00
Adobe!
```

Example 8

The **Host** and **Directory** values are the locations where the ARs are stored for this database. **Version** is a number which is incremented when the **Host**, **Directory**, or fieldlist is changed. **Version** is changed only when one of these aspects of the database is changed. **Next Submit Number** is the identifying number that will be assigned to the next AR submitted. **Start Time** and **Stop Time** are used by the updater process, which performs updates to the accelerator files. (The updater process was discussed in the previous section.)

Use **Examine Update List** to examine the updater process's list of ARs whose accelerator entries may need to be updated. Example 9 illustrates this command.

```
Adobe!Examine Update ListReturn
Adobe systems
 1. DataBase Name 1
 2. DataBase Name 2
 Enter System number (1 2): 1Return
  Update List is empty
Adobe!Examine Update ListReturn
Adobe systems
 1. DataBase Name 1
 2. DataBase Name 2
 Enter System number (1 2): 1Return
  Update List contains: 4-24, 50
Adobe!
```

Example 9

Simultaneous access to the accelerator files of a database is controlled. More than one client may read the accelerator files simultaneously, but a client who wishes to write into the accelerator file must have exclusive access. The way a database's accelerator files are being accessed at any moment may be examined with the **Show Accelerator Activity** command. In the first part of Example 10, one client is reading from the accelerator files. Another client is waiting for the first to complete so that it may write into the accelerator files. In the second example, a client is writing to the accelerator files.

```

Adobe!Show Accelerator ActivityReturn
Adobe systems
  1. DataBase Name 1
  2. DataBase Name 2
Enter System number (1 2): 1Return
Accelerator Reader Count: 1
Accelerator Status: writerWaiting
Adobe!Show Accelerator ActivityReturn
Adobe systems
  1. DataBase Name 1
  2. DataBase Name 2
Enter System number (1 2): 1Return
Accelerator Reader Count: 0
Accelerator Status: writerIn
Adobe!

```

Example 10

2.4.4 Changing the parameters of an Adobe database

Most of the system parameters of an Adobe database that are observable using the **ShowSystem Parameters** command can be altered by a system administrator. Some of these changes will cause the version number associated with that database to be incremented. The next time they attempt to perform any operation dependent on one of the changed system parameters or on the system description fieldlist, all client tools will detect the parameter or version number change. Then the clients must obtain updated information.

To set the next submit number, use the **Put Submit Number** command. This command is usually used in conjunction with the conversion or move (see 2.4.5) of a database. It has no effect on the database's version number. **Put Submit Number** is illustrated in Example 11.

```

Adobe!Put Submit NumberReturn
Adobe systems
  1. DataBase Name 1
  2. DataBase Name 2
Enter System number (1 2): 1Return
Enter Next Submit Number: (1..2147483647): 22Return
Confirm (Y/N): yRETURN
Adobe!

```

Example 11

To change the storage location of the database's ARs, use the **Change Location of ARs** command, shown in Example 12. Changing the location of ARs increments the database's version number.

```
Adobe!Change Location of ARsReturn
Adobe systems
  1. DataBase Name 1
  2. DataBase Name 2
Enter System number (1 2): 1Return
Enter New Host (host:domain:organization): FS3:Domain:
  OrganizationReturn
Enter New Directory: ARDrawer/DataBaseName1Return
Adobe!
```

Example 12

2.4.5 Moving an Adobe database

It may become necessary to move a database to a different Adobe service, usually because the Adobe service has become crowded; that is, when almost no space remains on the server for storage of accelerator files and other miscellaneous Adobe files.

To move an Adobe database from one Adobe service to another, follow these steps:

1. While the service is started, set the context to Adobe and invoke **Backup Systems**. Enter the number of the database you want to move. This command causes all the files associated with that database to be backed up.
2. Unregister the database by using the **UnRegister System** command.
3. Stop the Adobe Service from which you are removing the database. Do this by invoking the **Stop Service** command and then selecting the Adobe service.
4. Invoke the **Destroy AR System** command to remove all of the database's files from this Adobe service.
5. Restart this Adobe service
6. Start the Adobe service to which you want to move the database.
7. On the Adobe service to which you wish to move the database, invoke the command **Restore AR System**. You must supply this command with the name of the database to be restored and the location where its files are stored.
8. Register the database at its new location using the **Register System** command.
9. Stop and restart the Adobe service onto which you have moved the database.

2.4.6 Destroying an Adobe database

A database may need to be destroyed if a new database has superseded an old one, or if for some other reason a database is no longer needed. You may also want to destroy a *new* database that needs many changes and has few ARs submitted to it. (In this situation, it is easier to destroy the database you have set up, make all the needed changes to the system description file, and then put the database onto the Adobe service again.)

To destroy a database, follow these steps.

1. Stop the Adobe service on which the database is located.
2. Invoke the **Destroy AR System** command to remove all of the database's files from this Adobe service.
3. Unregister the database by using the **UnRegister System** command.
4. Restart the Adobe service.

2.4.7 Changing an Adobe database's fieldlist

As time goes by, you may need to modify the fieldlist of your database to better meet the needs of its users. The **Change Fieldlist** command is provided for this purpose. You may change the fieldlist by adding, removing, or modifying a component field. There are a number of ramifications in making changes to the database's fieldlist. You should understand these before you begin making changes.

Some fieldlist changes have more wide-reaching effects than others. The following chart gives the effects on accelerator files and the ARs themselves of each type of change to each type of field.

Change Type

Field Type:	Modify	Add	Remove
ARId	Rename accel file* Fix field name in all ARs*	Not allowed	Not allowed Destroy old accel file
Numeric	Rename accel file* Fix field name in all ARs*	Make new accel file	Remove field from all ARs
DateTime	Rename accel file* Fix field name in all ARs*	Make new accel file	Destroy old accel file Remove field from all ARs
BoundedString	Rename accel file* Recreate accel file** Fix field name in all ARs*	Make new accel file	Destroy old accel file Remove field from all ARs
UnboundedString	Fix field name in all ARs*	None	Remove field from all ARs
Enumerated	Rename accel file* Recreate accel file** Fix up ARs if choices change**	Make new accel file	Destroy old accel file Remove field from all ARs

*Necessary only if you change the field's name

**Necessary if you change the field's type-dependent attributes

Changes can only be made to only one field at a time. After you specify a change to be made and confirm it, all required work for updating the database to reflect this change

will be performed. You can determine just how much work you have given the service by looking up your change on the chart.

To use **Change Fieldlist**, you must be logged in and enabled. When you invoke the **Change Fieldlist** command, you will first be asked which Adobe system you want to change. After you have given the number corresponding to the system you wish to change, you will be asked what type of change you wish to make, **Modify**, **Add**, or **Remove**. Example 13 illustrates this first portion of interaction.

```
Adobe! Change FieldlistReturn
Adobe systems
 1. DataBase Name 1
 2. DataBase Name 2
 3. DataBase Name 3
 Enter System number (1 3): 1Return
Type of change?
 1. Modify
 2. Add
 3. Remove
 Enter number (1 3): 1Return
```

Example 13

2.4.7.1 Adding a field

After indicating that you wish to add a field, you will be asked to supply the type of the field to be added. The list of types you can choose from will be displayed; notice that you cannot add a field of type **ARId**. After selecting the field type, you will be prompted for all the attributes of the field. All field types will request a field name and a **SystemMust** default for each tool type. After you supply the field name, you are asked for any type-dependent attributes, e.g., the **MaxLength** for a **BoundedString** field or the list of **EnumeratedItems** for an **Enumerated**. The last values you must supply are for the **SystemMust** defaults. Merely press RETURN to indicate that there is no default. Example 14 shows the addition of an independent enumerated field. When all **EnumeratedItem** values have been given, press the RETURN key in response to the next **EnumeratedItem** prompt. If you were adding a dependent enumerated, you would be prompted for **EnumeratedItems** to go with each choice in the independent enumerated field on which your new field depends.

```

Type of change?
1. Modify
2. Add
3. Remove
Enter number (1 3): 2Return
Field Type?
1. DateTime
2. Numeric
3. Bounded String
4. Enumerated
5. Unbounded String
Enter choice number: 4Return
Field Name: NewListReturn
DependsOn?
2. EnumField1
10. EnumField2
15. EnumField3
16. EnumField4
20. EnumField5
Enter number (0 in NIL DependsOn): 0Return
Independent EnumeratedItems
EnumeratedItem: choice1Return
EnumeratedItem: choice2Return
EnumeratedItem: choice3Return
EnumeratedItem: choice4Return
EnumeratedItem: Return
SystemMust Default for Edit:Return
SystemMust Default for Submit:Return
SystemMust Default for Report:Return
SystemMust Default for Query:Return
SystemMust Default for Sort:Return
SystemMust Default for QueryList:Return
Is this right? (Y/N): YReturn
NewList added.

```

Adobe!

Example 14

2.4.7.2 Modifying a field

You can modify anything about any existing field except its type. Modifying a field is very much like adding a field, except that after you have selected the type of change desired, you are presented with a list of existing fields from which to select the field you want to modify. Each attribute of the field is then displayed for you with its current value filled in. To leave a value as it is, simply press the RETURN key. Type in the new value if you want to change the attribute. You should be aware that, as indicated by the chart presented earlier, changing the name of a field or its type-dependent attributes results in a lot of work to be done to update the existing database to fit the new description. The Adobe service does all the necessary work for you; however, it takes quite a bit of time for a sizeable database to be processed, so be prepared. Feedback is given by the service to indicate how it is progressing in its efforts to make the database consistent with the new description. Note that if you change the length of a **BoundedString** field or remove enumerated items from an enumerated field, that field's accelerator file must be

recreated. Example 15 shows a **BoundedString** field, **ShortString**, having its name changed to **ShorterString** and **MaxLength** changed from 40 to 30. This example assumes the database has 379 ARs.

```
Type of change?
1. Modify
2. Add
3. Remove
Enter number (1..3): 1Return
Which field?
1. Number
2. Date
3. Subject
4. Status
5. Priority
6. Originator
7. Description
8. ShortString
Enter number (1..8): 8Return
FieldName: ShorterString Return
MaxLength (1..512): 30 Return
SystemMust Default for Edit: Return
SystemMust Default for Submit: Return
SystemMust Default for Report: Return
SystemMust Default for Query: Return
SystemMust Default for Sort: Return
SystemMust Default for QueryList: Return
Is this right? (Y/N): y Return
Editing ARs: 1..20..40..60..80..100..120..140..160..180..200..220..240..260..
280..300..320..340..360..379
Making Accelerator File for ShorterString: 1..20..40..60..80..100..120..140..160..180..
200..220..240..260..280..300..320..340..360..379
    ShorterString modified.
Adobe!
```

Example 15

2.4.7.3 Removing a field

Any field is a candidate for removal from the fieldlist except the ARId field. Anytime you remove a field, all ARs in the database must be edited by the service to remove that field from each AR. Example 16 illustrates the deletion of a field. This example assumes the database has 379 ARs.

```
Type of change?
 1. Modify
 2. Add
 3. Remove
 Enter number (1..3): 3Return
Which field?
 1. Number
 2. Date
 3. Subject
 4. Statue
 5. Priority
 6. Description
 7. Other
 Enter number (1..7): 2Return
 Remove Date? (Y/N): yReturn
Editing ARs: 1..20..40..60..80..100..120..140..160..180..200..220..240..260..
280..300..320..340..360..379
Date removed.
Adobe!
```

Example 16



Librarian service

This document describes how to set up and maintain your Librarian service and its data bases. The last section gives an index of all Librarian service commands.

3.1 Introduction

The Librarian service can be thought of as a simple database service. Each Librarian supports one or more databases of objects known as libjects. A libject is a named property list residing within some database. Each of the properties on that list is tagged with the name. The librarian supports applications that do simple queries and edits on libjects.

The actual properties contained within each libject are determined by the applications themselves. For example, the LibrarianTool and DF software create and manipulate libjects associated with implementor's files. These tools use the librarian to prevent simultaneous editing of their files and to indicate where their files are. The libjects in these type of data bases contain properties such as **checkedOut**, **checkedOutByWhom**, and **remoteFileName**.

Other applications may use the librarian for entirely different purposes. Since there is no type associated with a data base, an application must know the name of the data base it wishes to access.

3.2 Setting up your Librarian service

3.2.1 Installing a Librarian service

Before installing your Librarian service, it is important that you first read the *Product Descriptions* booklet, the "Services Executive" section in the *Server Operation and Maintenance* booklet, and the *Introduction to Network System Administration* booklet. You must also perform the procedures described in the *Server Software Installation* booklet. These booklets are part of the *Basic Network Information* volume.

The files needed for installing your Librarian service, **LibrarianService.bcd** and **LibrarianSDF.bcd**, can be found on the release directory. Like all services, you can either

install the librarian from a floppy, or you may retrieve these files directly to the server and then activate and run the service explicitly.

Once the librarian service has been run, its set of commands will be available through the executive. The prompt for the librarian is **LS>**. Commands to examine the service and its data bases require that you be logged on. Commands which modify a data base or its parameters require that you be logged on and enabled.

3.2.2 Creating a new database

Creating a Librarian database is a two-step process. First you must create the associated files on the service using the **Create Database** command. This will create the **.Records** file, which contains the libjects, the **.BTree** file, an index into the **.Records** file, and the **.Log** file, which maintains a log of transactions.

The next step is to register the name of your database with the Clearinghouse using the **Register Database** command. This associates the database name with the name of the service on which it lives, allowing Librarian tools to connect to the appropriate server. The database will be registered in the domain and organization of this server unless otherwise specified.

Example:

```
LS!Create Database
Database name: MesaDFs
Creating 'MesaDFs' ... done.
LS!Register Database
Database name: MesaDFs
Registering 'MesaDFs:OSBU North:Xerox' ... done.
```

3.2.3 Database parameters

Certain parameters associated with each database should be set after the database has been created.

Set Authentication Level sets the flavor of authentication necessary to connect to the database. The choices include **simple** and **strong**, with **simple** being the default. Note: Only simple authentication is supported at this time.

Set Backup Path sets the remote directory location of the database files to be backed up. The syntax is **(host name:domain:organization)File drawer/dir₁/.../dir_n**. The domain and organization must be included. Backup and recovery will be discussed in the next section.

Set Readers sets the list of readers for that database. These are the people that may query or examine the database. **Set Writers** sets the list of writers. These are people who may update or modify the database. The arguments must either be Clearinghouse groups or individuals. If neither of the two parameters is set, then everyone is allowed read and write access.

3.2.4 Listing and tracing

List Databases lists those databases matching the specified pattern. Answering **Yes** to **Verbose feedback?** lists the associated database parameters, and the **.Records**, **.BTree**, and **.Log** files with their last modified dates and size in bytes.

Toggle Tracing flips the tracing switch between on and off. When tracing is on, messages indicating opening or closing of sessions will be posted at the service along with the initiator's name. These messages are suppressed when tracing is off. The service starts up with tracing on.

3.3 Maintaining your Librarian service

3.3.1 Backup and recovery

Backup for each database on the service occurs automatically every 24 hours. At this interval, the Librarian stores out the **.Records** and **.BTree** files to the remote backup location specified by the **Set Backup Path** command. If for some reason the backup fails, for instance, if the file service is unavailable, the Librarian will try again at half-hour intervals until it succeeds.

After storing each file, the Librarian deletes all but the three highest versions. For backup to be successfully completed, the Librarian service itself must have add and remove access to the remote backup directory. The Librarian uses the service credentials, not the server or logged-on user, to access the remote directory, so use the name of the Librarian service as the entry in the file access list.

There is also an explicit **Backup Databases** command if you do not want to wait for the automatic backup process to awaken. This command is useful for moving a database or a Librarian service.

A database will need to be recovered if the server crashes in the middle of a transaction that attempts to update the database. After rebooting the server, a load operation for the damaged database will fail with the message

LS! MesaDFs: needs recovery.

If this occurs, you must recover the database explicitly using the **Recover Database** command. It will prompt you for the database to be recovered and the recovery path. The current backup path will be filled in for recovery path, so in most cases you can simply press **RETURN**.

Recover Databases deletes the existing **.Records** and **.BTree** files on the disk and then retrieves the latest ones from the backup location. The Librarian warns that the existing database will be deleted and asks for confirmation. You should be certain a reasonable backup exists before executing this command. It then plays the log of transactions that have occurred since the last backup. These transactions are kept in the **.Log** file. If this file becomes damaged, you could lose a record of transactions that have occurred since the last backup.

Future enhancement: The Librarian will back up its log files every half hour while continuing to back up its **Records** and **BTree** files every day, to minimize the impact of a smashed log file. It will also recover databases automatically as necessary.

3.3.2 Deleting and moving databases

To delete a database, first execute the **Destroy Database** command. This deletes all data files associated with the database. You must be logged on and enabled, and the service must be stopped. Then remove the database entry from the Clearinghouse using the **Unregister Database** command. This also removes its name from the group of database names the service claims to support.

You may want to move a database if it has gotten too big for the server's local disk. You should try to avoid this situation and anticipate the maximum space your database will need. However, if this or some other situation arises, use the following steps to move a database.

First stop the service to prevent any further transactions from taking place until the database has been moved. Then explicitly back up the database using the **Backup Databases** command. When you are certain a successful backup has completed, delete the database as described in the previous paragraph, using the **Destroy Database** and **Unregister Database** commands.

Now at the destination Librarian service, create a database of the same name using the **Create Database** and **Register Database** commands. Clearinghouse propagation delays may prevent you from registering the database with its new service right away. Finally, execute the **Recover Database** command, specifying the previous backup location.

Note: The **LibrarianStub** resident on the workstation maintains a cache of database locations. If a database or Librarian service is moved, workstations must be rebooted to discover this new location.

3.4 Librarian service command index

This section lists all the Librarian service commands in alphabetical order, followed by a brief explanation. To address the Librarian service commands, you must be in the Librarian service context.

Backup Databases

Starts the backup process on all loaded databases.

Create Database

Creates a new Librarian database with the given name. Does *not* register the database with the Clearinghouse.

Destroy Database

Available to the enabled user when the Librarian service is stopped. Does *not* unregister the database from the Clearinghouse.

List Databases

Enumerates the specified databases and their corresponding parameters and data files.

Recover Database

Recovers the specified database from the backup location.

Register Database

Registers the specified database name with the Clearinghouse.

Set Authentication Level

Sets the flavor of authentication to either **simple** or **strong** for the specified database.

Set Backup Path

Sets the remote backup location for the specified database.

Set Readers

Sets the group or individuals allowed to read the specified database.

Set Writers

Sets the group or individuals allowed to perform transactions that write to the specified database.

Toggle Tracing

Flips the tracing switch between on and off.

Unregister Database

Removes the specified database name from the clearinghouse and also removes it from its Librarian services's Clearinghouse group.

3-6