Digital X.500 Directory Synchronizer Release Notes V1.3-2

Software Version: V1.3-2

Date: April 1996

Digital Equipment Corporation

Maynard, Massachusetts

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with terms of such license.

No responsibility is assumed for the use or reliability of equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013."

Copyright ©1996 by Digital Equipment Corporation.

All rights reserved.

Printed in U.S.A.

Trademarks

The following are trademarks of Digital Equipment Corporation: AXP, DEC, DECnet, Digital, OpenVMS, VAX, VMS and the DIGITAL logo.

The following are trademarks of other companies: OSF and OSF/1 are registered trademarks of Open Software Foundation, Inc. OSI is a registered trademark of CA Management, Inc. UNIX is a registered trademark in the United States and other contries licensed exclusively through X/Open Company Ltd.

Contents

Chapter 1 V1.3-2 New Features

1.1 New Syntaxes Supported 1–1
1.2 Allow modification of UNIQUE_FIELD if MAKE_UNIQUE=NO 1–1
1.3 Separate RDF files for TRANSACTION mode processing 1–2
1.4 Improved X.500 error reporting 1–2

Chapter 2 Problems fixed in V1.3-2

2.1 PostalAddressSyntax not allowing T.61 characters 2–1
2.2 Error in RDF processing when column=* 2–1
2.3 Modify in TRANSACTION Mode not checking synchid, gid 2–1
2.4 UPCASE not working with international characters 2–1

Chapter 3 V1.3-2 Restrictions

3.1 XDSU_CONFIG.DAT SYNCH_OBJECT in IMPORT mode 3–1
3.2 Supported Syntaxes 3–1
3.3 Teletex attribute constraint violation problem 3–2

Chapter 4 V1.3-1 New Features

4.1 MOD_OR_ADD_FLAG for TRANSACTION Mode 4–1
4.2 Boolean Syntax 4–1
4.3 RDF Reference attribute 4–2
4.4 Special Flag to store attributes in Upper Case 4–2

Chapter 5 Problems fixed in V1.3-1

5.1 XDSU-F-MEMALLOC error if all unique check flags removed 5–1
5.2 TRANSACTION mode UNIQUE_FIELD Processing 5–1
5.3 "Unknown Code" error status 5–1
5.4 Problem with RDF comment lines ending in "-" 5–1
5.5 ADMD_SPACE not working 5–2
5.6 Error deleting entry when CN contains a "/" or "," 5–2
5.7 Removing a value in a multi-value attribute modification not seen 5–2

Chapter 6 Mandatory Changes to RDF files for V1.3

6.1 Double Quote Characters in RDF constant strings 6–1

Chapter 7 V1.3 New Features

7.1 Bind to non-default Presentation Address 7–1
7.2 Passwords and Access Control 7–1
7.3 Enhanced RDF conditional logic 7–2
7.4 Support for MTS Distribution Lists 7–3
7.5 Check for Duplicate Values in a multi-valued attribute 7–5
7.6 New Warning Messages when X.500 attributes not found in Layout File 7–5
7.7 Multipart RDN support 7–6

Chapter 8 Problems fixed in V1.3

8.1 SORTFAILED Error on OpenVMS 8–1
8.2 Problem with double quotes in RDF Export Strings 8–1
8.3 DELETE of existing entries fail after changing DN_SEQUENCE 8–1
8.4 XDSU terminating after XDSU-W-BADVALUE message 8–1
8.5 XDSU V1.2 Spaces in UNIQUE_FIELD_SEPARATOR Problem 8–2
8.6 XDSU-F-X500FETCH error and ENCODE_DELIM 8–2
8.7 distinguishedName Syntax Problems 8–2

Chapter 1
V1.3-2 New Features

1.1 New Syntaxes Supported

For V1.3-2, the following X.500 Syntaxes are now supported by XDSU. They are listed with the valid ascii input format in the foreign directory file:

- bitStringSyntax - string of ascii 0's and 1's.
- generalizedTimeSyntax - a generalized time string like 19960415140545.000Z which represents 1996, April 15th, at 2:05:45pm.
- integerSyntax - a valid ascii integer string.
- octetStringSyntax - a string of 2-digit hex numbers in ascii like A1B2C3.
- telephoneNumberSyntax - a printable string.
- undefinedSyntax - the verbatim ASN.1 encoded string of 2-digit hex numbers.
- userPasswordSyntax - a printable string.
- utcTimeSyntax - a UTC time string like 960415140545Z which represents 1996, April 15th, at 2:05:45pm.

1.2 Allow modification of UNIQUE_FIELD if MAKE_UNIQUE=NO

For V1.3-2, in IMPORT or TRANSACTION mode, the UNIQUE_FIELD may be modified if MAKE_UNIQUE=NO.

Prior to V1.3-2, XDSU would not recognize modification of the UNIQUE_FIELD attribute in existing records during IMPORT or TRANSACTION modes. If MAKE_UNIQUE=YES this makes sense since the value could be different every time an import is done, if it needs to be made unique.

However, if MAKE_UNIQUE=NO, this attribute should be allowed to change - if the new attribute value is still unique.

1.3 Separate RDF files for TRANSACTION mode processing

For V1.3-2, in TRANSACTION mode, three separate RDF files - one for each input file may now be optionally specified.

Prior to V1.3-1, XDSU TRANSACTION mode only allowed one RDF file for delete, modify, and add processing. So, if the format of the delete, modify, or add file was different from another, multiple XDSU TRANSACTION runs had to be setup to process one input file at a time.

Support for one RDF file is still maintained - and is the default. This is specified by the "rdf_file" configuration keyword. But if "rdf_file" is not specified - then the add, modify, or delete specific RDF file is used ("add_rdf_file", "mod_rdf_file", "del_rdf_file").

1.4 Improved X.500 error reporting

For V1.3-2, the actual X.500 error is logged when an X.500 error occurs.

Chapter 2

Problems fixed in V1.3-2

2.1 PostalAddressSyntax not allowing T.61 characters

For V1.3-2, PostalAddressSyntax will accept teletex (T.61) string characters instead of only printable string characters.

2.2 Error in RDF processing when column=*

For V1.3-2, during EXPORT or CHANGES RDF processing, an error that was occuring when the first reference to a row# > 1 has a column# = '*' has been fixed. This was causing the output to go at the end of the previous line. It will now go on the beginning of the correct line.

2.3 Modify in TRANSACTION Mode not checking synchid, gid

In XDSU import TRANSACTION Mode, when doing a modify, XDSU was not checking the synch id and group id of the X.500 record being modified to see that it matched the input record. In V1.3-2, if the synch id and group id's don't match then XDSU will not modify X.500 and fail the modify transaction.

2.4 UPCASE not working with international characters

In V1.3-2, UPCASE will also cause international characters to be in Uppercase.

Chapter 3
V1.3-2 Restrictions

3.1 XDSU_CONFIG.DAT SYNCH_OBJECT in IMPORT mode

If you change SYNCH_OBJECT after an inital load, existing entries' objectClass will NOT be modified. It is important to set SYNCH_OBJECT correctly the first time. Workarounds include deleting all entries and reloading or using DXIM to update the objectClass attribute:

```
dxim> Select <search_context> where <filters> and commonname=*
dxim> modify selected add value objectclass=(<new class>)
```

3.2 Supported Syntaxes

XDSU can access attributes having the following syntaxes:

```
bitStringSyntax
booleanSyntax
distinguishedNameSyntax
facsimileTelephoneNumberSyntax
generalizedTimeSyntax
ia5StringSyntax
integerListSyntax
integerSyntax
mhs-or-address-Syntax
mhs-or-name-Syntax
numericStringSyntax
objectIdentifierSyntax
octetStringSyntax
postalAddressSyntax
printableStringSyntax
stringListSyntax
stringSyntax  (defined as printableStringSyntax or teletexStringSyntax)
telephoneNumberSyntax
teletexStringSyntax
teletexTerminalIdentifierSyntax
telexNumberSyntax
undefinedSyntax
userPasswordSyntax
utcTimeSyntax
```

3.3 Teletex attribute constraint violation problem

A teletexStingSyntax attribute cannot contain the maximum number of characters for a field if it also contains certain international characters (e.g. é and è) which take up 2 characters of ASN.1 storage.

Chapter 4

V1.3-1 New Features

4.1 MOD_OR_ADD_FLAG for TRANSACTION Mode

For V1.3-1, a new optional XDSU_CONFIG.DAT parameter ("MOD_OR_ADD_FLAG") has been added for import TRANSACTION Mode (SYNCH_TYPE=TRANSACTION). This parameter can have a value of YES or NO. If YES, an attempt to modify entries in the INPUT_MOD_FILE is made, but if the entry does not already exist in X.500, the entry is added. If NO, the transaction will fail if the entry does not already exist.

This is useful if the foriegn directory feed into XDSU cannot distinguish between adds and modifies. So all adds and modifies go in the same data file which can now be specified as the INPUT_MOD_FILE and the MOD_OR_ADD_FLAG is set to YES.

This parameter is optional and defaults to NO.

4.2 Boolean Syntax

For V1.3-1, the X.500 booleanSyntax is supported by XDSU.

When storing a booleanSyntax attribute on import, if the value for the attribute begins with a 'T', 'Y', or '1', a TRUE boolean value is stored. Otherwise a FALSE boolean value is stored. For example in an import RDF:

```
XXXenable = "YES"     ! sets the value of this Boolean attribute to TRUE
```

On export, a TRUE boolean attribute is converted to a "1" in the output text file and a FALSE boolean attribute is converted to a "0". For example in an export RDF:

```
? XXXenable == "1" : "Boolean attribute is TRUE" = 1.1.#
```

4.3 RDF Reference attribute

In V1.3-1, an RDF "reference" syntax has been implemented for EXPORT mode. This syntax allows the export to use an distinguishedNameSyntax attribute for an entry to read ANOTHER entry and return an attribute from that second entry.

The RDF syntax is:

```
dn_attribute->ref_attribute
```

For example, if you wanted to return the mhs-or-addresses attribute of the dn specified in the seeAlso field:

```
seeAlso->mhs-or-addresses = 1.1.256
```

In this case, the seeAlso field value is used to read another X.500 entry from which the mhs-or-addresses attribute is returned.

So if:

```
/c=us/o=digital/cn=megan
 seeAlso = /c=us/o=digital=cn=michelle

/c=us/o=digital/cn=michelle
 mhs-or-addresses = c=us; a=a; p=d; o=d; ou1=family; cn=michelle

   an export of "/c=us/o=digital/cn=megan", "seeAlso->mhs-or-addresses" would
   return "c=us; a=a; p=d; o=d; ou1=family; cn=michelle"
```

The "dn_attribute" has to be distinguishedNameSyntax and this is only supported for EXPORT mode. (not CHANGES).

4.4 Special Flag to store attributes in Upper Case

A special flag has been added to force attributes to be stored in Upper Case in X.500 on IMPORT. Although X.500 is not case sensitive, a foriegn directory or database getting an export from X.500 might be. To a force an attribute to be stored in uppercase, add a special flag called UPCASE (which would go in field 9 in the layout file for an ATTR record).

For example, to force the title field to be stored in X.500 in uppercase:

```
ATTR|title|64|{2 5 4 12}|teletexStringSyntax|0|4||UPCASE|
```

Chapter 5
Problems fixed in V1.3-1

5.1 XDSU-F-MEMALLOC error if all unique check flags removed

If all unique check flags are removed from the XDSU_LAYOUT.DAT file, an XDSU-F-MEMALLOC fatal error was returned on Digital UNIX (DEC OSF/1) versions of XDSU. (The unique check flags are defined in Field 8 in the layout file for ATTR records.)

This has been fixed in XDSU V1.3-1.

5.2 TRANSACTION mode UNIQUE_FIELD Processing

For import TRANSACTION mode, during MODIFY processing, the UNIQUE_FIELD attribute was being modified. It should not be modified during MODIFY processing, but only set during ADD processing. The same is true for the PRU and SNU fields if GEN_PROFS_ALIAS and GEN_SNADS_ALIAS are YES.

This has been fixed in XDSU V1.3-1.

5.3 "Unknown Code" error status

A syntax error in adding or modifying mhs-or-address syntax or mhs-or-name syntax attributes resulted in an "status=Unknown Code" instead of "status=BADFLDVAL" in the XDSU transaction log file.

This has been fixed in XDSU V1.3-1.

5.4 Problem with RDF comment lines ending in "-"

In an RDF file, a comment line which ended in a hyphen (the line continuation character) caused the next line to be treated as a comment.

This has been fixed in XDSU V1.3-1.

5.5 ADMD_SPACE not working

The ADMD_SPACE import configuration parameter, which if =YES forces the ADMD attribute to be stored containing a single space, was not working. Also, when using distribution lists, the XDSU_LAYOUT_MTS.DAT file should have the ADMD flag set for the mts-ADMD-Name attribute.

This has been fixed in XDSU V1.3-1.

5.6 Error deleting entry when CN contains a "/" or ","

XDSU gets an error Deleting an entry where the last RDN in the DN (usually commonName) contains a "/" or "," character.

This has been fixed in XDSU V1.3-1.

5.7 Removing a value in a multi-value attribute modification not seen

XDSU did not modify a multi-value attribute if one of its existing values was removed on an Import. The modify did occur if an existing value is changed or added.

This has been fixed in XDSU V1.3-1.

Chapter 6
Mandatory Changes to RDF files for V1.3

6.1 Double Quote Characters in RDF constant strings

As discussed in detail in the following V1.3 New Features section, conditional logic support in the RDF files has been enhanced from earlier versions. Due to the increased parsing complexity of this change, constant character strings defined in the RDF must be changed if they include a double quote character. To include a double quote character in a constant string, two consectutive double quote characters must now be supplied to produce one double quote character.

For example, for an import RDF:

```
mhs-or-addresses="; CN=""
mhs-or-addresses=1.1.10
mhs-or-addresses=" "
mhs-or-addresses=1.11.10
mhs-or-addresses=""
```

```
    must be changed to
```

```
mhs-or-addresses="; CN="""
mhs-or-addresses=1.1.10
mhs-or-addresses=" "
mhs-or-addresses=1.11.10
mhs-or-addresses="""
```

And for an export RDF:

```
"surname = ""=1.1.#
surname=1.*.#
"""=1.*.#
```

```
    must be changed to
```

```
"surname = """=1.1.#
surname=1.*.#
""""=1.*.#
```

Chapter 7
V1.3 New Features

7.1 Bind to non-default Presentation Address

XDSU V1.3 supports binding to X.500 using a specified DSA presentation address. The previously unimplemented XDSU_CONFIG.DAT parameter, BIND_DSA, will be used to contain this address. The format of the bind_dsa parameter will the same as the DUA.KnownDSAs.paddr parameter in the dua defaults file. For example:

```
bind_dsa="DSA"/"DSA"/"DSA"/NS+49003DBB000400ADF421,CLNS
```

This parameter will be optional and the value specified in the dua defaults file will be used if not specified.

7.2 Passwords and Access Control

XDSU V1.3 supports binding to X.500 using a requestor name and password. This allows XDSU to access X.500 data which has been placed under access control and may require a password.

There are two new optional XDSU_CONFIG.DAT parameters:

```
bind_name= /C=US/O=DIGITAL/CN=SECURITY
bind_password= SECURITYPASSWORD
```

These attributes correspond to the dxim -c bind parameters: name and password. Since the password does appear in the xdsu_config.dat file, proper protection of this file should be considered. The password, however, will not be printed out to the log file or included in any mail messages.

7.3 Enhanced RDF conditional logic

XDSU V1.3 contains support for enhanced RDF conditional logic. Previous versions of XDSU only contained support for RDF conditionals which allow logic based on the presence or absence of an attribute:

```
?field:        ! if field is non-blank
~field:        ! if field is blank (or not-present)
```

The new logic will allow attributes to be compared with string literals or other attributes using EQUAL (==), NOT EQUAL (!=), SUBSTRING OF (<<), and NOT A SUBSTRING OF (!<) operators:

```
? field  == "STRING" :      ! True if field equals STRING
? field1 !=  field2  :      ! True if field1 does not equal field2
? field  << "STRING" :      ! True if field is a substring of STRING
? field  !< "STRING" :      ! True if field is not a substring of STRING
```

All comparisons are case insensitive (e.g. "a" equals "A").

Comparisons can be combined by use of logical AND (&&) and OR (||) operators:

```
? field == "A" || field == "B" :    ! True if field equals A or B
? field != "A" && field != "B" :    ! True if field not equal to A and B
? field << "ABCD" || field = "E" :  ! True if field is a substring of
                                    !  ABCD or field equals E
```

Parenthesis can also be used for complex conditionals:

```
? (field == "A" || (field != "B" && field1 != field2) ) :
```

The boolean NOT (!) operator can also be used:

```
? !(field == "A" || field == "B") :  ! True if field is not A or B
? !field :                           ! True if field is blank
                                     !  (Same as ~field: )
```

Also, an optional ELSE conditional statement after a second colon (':') has been implemented:

```
? 1.1.10 : description = 1.1.10 : description = "Default"
```

instead of having to do:

```
? 1.1.10 : description = 1.1.10
~ 1.1.10 : description = "Default"
```

So the format of an RDF conditional is:

```
[ {? | ~} <conditional> : ] <field> = <field> [ : <field> = <field> ]
```

And, since RDF lines can now be pretty big, lines may be continued on the next line by ending a line with the hypen ('-') character:

```
?(<field1> == "TEST") || (<field2> != "ORG1" && <field2> !="ORG2"): -
    organization = "ABCD" : -
    organization = "EFGH"
```

7.4 Support for MTS Distribution Lists

XDSU V1.3 contains support for importing/exporting MTS Distribution Lists to/from X.500. This allows distribution lists to be managed from a simple text file using XDSU instead of using NCL scripts.

To support MTS Distribution Lists, the mhs-or-name-syntax has now been implemented. In addition, a sample XDSU Layout File (xdsu_layout_mts.dat) has been included in XDSU V1.3 and may be found in the XDSU root directory. This file must be specified on the XDSU command line as the second parameter. This file contains a sample of the Object Classes, RDN, and Attribute definitions needed for MTS Distribution List entries and any parent entries.

Sample Files for MTS Distributon list support:

```
XDSU_LAYOUT_MTS.DAT
-------------------------------------------------------------------------------
OBJ|dec-mts-distribution-list|{1 3 12 2 1011 5 4 2 22}

RDN|distinguishedName|512|000000|00|1|0||DNAM|000000
RDN|country|2|{2 5 4 6}|printableStringSyntax|1|0|||{2 5 6 2}
RDN|organization|64|{2 5 4 10}|printableStringSyntax|1|0|||{2 5 6 4}
RDN|MTS|64|{1 3 12 2 1011 5 4 1 1}|printableStringSyntax|1|0|||
                                                    {1 3 12 2 1011 5 4 2 0}
RDN|MTS-CountryName|2|{1 3 12 2 1011 5 4 1 27}|printableStringSyntax|1|0|||
                                                    {1 3 12 2 1011 5 4 2 5}
RDN|MTS-ADMD-Name|64|{1 3 12 2 1011 5 4 1 28}|printableStringSyntax|1|0|||
                                                    {1 3 12 2 1011 5 4 2 6}
RDN|MTS-PRMD-Name|64|{1 3 12 2 1011 5 4 1 29}|printableStringSyntax|1|0|||
                                                    {1 3 12 2 1011 5 4 2 7}
RDN|dec-mts-organization-name|64|{1 3 12 2 1011 5 4 1 39}|printableStringSyntax
                                                |1|0|||{1 3 12 2 1011 5 4 2 8}
RDN|dec-mts-organizational-unit-name|64|{1 3 12 2 1011 5 4 1 40}|printableString
                                              Syntax|4|0|||{1 3 12 2 1011 5 4 2 9}
RDN|dec-mts-common-name|64|{1 3 12 2 1011 5 4 1 38}|teletexStringSyntax|1|3||
                                                                  DVAL|000000
!
! Change 100 to the max# of members per distribution list for your system
!
ATTR|mhs-dl-members|256|{2 6 5 2 3}|mhs-or-name-syntax|0|100|||
ATTR|description|1024|{2 5 4 13}|printableStringSyntax|0|1|||
-------------------------------------------------------------------------------
```

```
Sample XDSU_CONFIG.DAT to import the dist lists:
-------------------------------------------------------------------------------
synch_type=IMPORT
synch_label=MTS_TEST
!
! Files and Run Options
!
rdf_file=MTS_TEST.RDF
input_for_dir_file=MTS_TEST.TXT
nrec_subscr=5
log_file=MTS_TEST.LOG
initial_load=NO
!
! Data Options
!
dn_sequence=/C/O/MTS/MTS-CountryName/MTS-ADMD-Name/MTS-PRMD-Name
            /dec-mts-organization-name/dec-mts-common-name
create_root=MTS-CountryName
synch_object=dec-mts-distribution-list
synch_id=MTS-TEST
synch_id_field=DESCRIPTION
gid_match=MTS-TEST
gid_field=DESCRIPTION
-------------------------------------------------------------------------------

Sample RDF which only assumes a max of 5 entries per Dist List:
-------------------------------------------------------------------------------
country="US"
organization="Digital"
MTS="Digital"
MTS-CountryName="US"
MTS-ADMD-Name="NE"
MTS-PRMD-Name="Digital"
dec-mts-organization-name="Digital"
!
dec-mts-common-name=1.1.20
mhs-dl-members   = 1.21.60
mhs-dl-members_2 = 2.21.60
mhs-dl-members_3 = 3.21.60
mhs-dl-members_4 = 4.21.60
mhs-dl-members_5 = 5.21.60
-------------------------------------------------------------------------------

Sample Input File:
-------------------------------------------------------------------------------
DIST-LIST1          C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Washington
                    C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Adams
                    C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Jefferson
                    C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Madison
                    C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Monroe
DIST-LIST2          C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Adams
                    C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Jackson
                    C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Van Buren

-------------------------------------------------------------------------------
```

```
This would create two entries that looks like this:
------------------------------------------------------------------------------
    /C=US/O=digital/MTS=Digital/MTS-CountryName=US/MTS-ADMD-Name=NE -
  /MTS-PRMD-Name=Digital/dec-mts-organization-name=Digital -
  /dec-mts-common-name=DIST-LIST1
      dec-mts-common-name = DIST-LIST1
      Object Class   = MTS Distribution List
      Description    = MTS-TEST
      Distribution List Member
  = C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Washington
  = C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Adams
  = C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Jefferson
  = C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Madison
  = C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Monroe

    /C=US/O=digital/MTS=Digital/MTS-CountryName=US/MTS-ADMD-Name=NE -
  /MTS-PRMD-Name=Digital/dec-mts-organization-name=Digital -
  /dec-mts-common-name=DIST-LIST2
      dec-mts-common-name = DIST-LIST2
      Object Class   = MTS Distribution List
      Description    = MTS-TEST
      Distribution List Member
  = C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Adams
  = C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Jackson
  = C=US; A=NE; P=Digital; O=Digital; OU1=COP; CN=Van Buren
------------------------------------------------------------------------------
Parent entries (e.g. /C=US/O=digital/MTS=Digital/MTS-CountryName=US) would
also be created.
```

7.5 Check for Duplicate Values in a multi-valued attribute

XDSU V1.3 performs duplicate value filtering. Previous versions of XDSU would give an error if a multi-valued attribute contained the same value more than once. In V1.3, XDSU issues a warning message (%XDSU-W-DUPVALDIS), and discards the duplicate values for the attribute.

7.6 New Warning Messages when X.500 attributes not found in Layout File

When XDSU is extracting entries from X.500 and the entry contains an RDN or attribute whose Object Identifier (OID) is not defined in the XDSU_LAYOUT.DAT (X.500 schema) file, it is no longer ignored.

Two new warning messages have been added:

```
%XDSU-W-OIDRDNNOTFND, Error finding RDN OID {n n n} in layout file
                      after <matched part of DN>

%XDSU-W-OIDATTRNOTFND, Error finding ATTR OID {n n n} in layout file
                       for <DN>
```

This alerts the XDSU Directory Administrator that the XDSU_LAYOUT.DAT file may need to be updated to match the X.500 schema.

7.7 Multipart RDN support

XDSU V1.3 supports use of multi-part RDN's. For example, as described in the X.500 Management manual, the following is a DN with a multipart RDN.

```
/c=US/o=Abacus/ou=Settlements,cn="Alfred Shaw"
```

To use multipart RDN's, for IMPORT and TRANSACTION mode operations, the multiple parts of a multipart RDN must be separated by a comma (',') in the DN_SEQUENCE config parameter:

```
DN_SEQUENCE=/C/O/OU,CN
```

Each RDN piece of the multi-part RDN must be specified in the XDSU Layout (schema) file as a separate entry and may contain only one RDN value (in column 6).

Also, attributes with the distinguishedNameSyntax (e.g. seeAlso) will now also support multipart RDN's.

Chapter 8
Problems fixed in V1.3

8.1 SORTFAILED Error on OpenVMS

On OpenVMS versions of XDSU only, during meta file sorting after an extract from X.500, an error in the sort command syntax was causing too much temporary disk space to be used during the sort. This was causing a SORTFAILED error if sort runs out of disk space and caused XDSU to abort.

This has been fixed in XDSU V1.3.

8.2 Problem with double quotes in RDF Export Strings

During EXPORT and CHANGES mode, specifying an EXPORT_STRING containing a double quote (") character in the RDF resulted in an INVRDFREC error. Specifying two double quote characters in a row ("") removed the error but both characters were output to the output file when only one was expected.

This has been fixed in XDSU V1.3 and due to the enhanced RDF conditional support, two double quote characters in a row must be specified to output a single double quote character to the output file.

8.3 DELETE of existing entries fail after changing DN_SEQUENCE

XDSU was incorrectly assuming the record to be deleted has the same DN_SEQUENCE as in the config file. Therefore, deleting of existing entries after changing the DN_SEQUENCE failed.

This has been fixed in XDSU V1.3.

8.4 XDSU terminating after XDSU-W-BADVALUE message

XDSU was terminating after receiving a %XDSU-W-BADVALUE warning message.

This has been fixed in XDSU V1.3.

8.5 XDSU V1.2 Spaces in UNIQUE_FIELD_SEPARATOR Problem

XDSU failed to remove the double quotes from around the UNIQUE_FIELD_SEPARATOR keyword value during unique field processing.

This has been fixed in XDSU V1.3.

8.6 XDSU-F-X500FETCH error and ENCODE_DELIM

For teletexStringSyntax attribute values, XDSU failed to encode the ENCODE_DELIM character so that any ENCODE_DELIM characters in the input stream would be improperly decoded when extracted from X.500. If the ENCODE_DELIM character was in an RDN value, this resulted in an XDSU-F-X500FETCH error.

This has been fixed in XDSU V1.3. Note that the ENCODE_DELIM configuration value need only be used if the {, }, ©, ¹, or \ characters are used.

8.7 distinguishedName Syntax Problems

XDSU failed to handle a DN string which had an RDN value surrounded by double quotes - which are needed if the RDN Value contains a "/" or ",".

This has been fixed in XDSU V1.3.