Digital X.500 Directory Synchronizer Release Notes V1.4

Software Version: V1.4

Date: October 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 V1.4 New Features

1.1 Limited Use License Support Added 1–1
1.2 CHANGES Mode Optionally Compares X.500 against Foreign Directory 1–1
1.3 Case Changes Optionally Recognized 1–2
1.4 More Detail Logs for CHANGES mode Added 1–2
1.5 Substring filter ability for GET_VALUE added 1–2
1.6 MHS-OR-ADDRESS Syntax Comparision Restriction Removed 1–3
1.7 Script Files Added to make running XDSU easier 1–3
1.8 New XDSU subdirectories added 1–3
1.9 XDSU_CONFIG_INSTANCE procedure improved 1–4
1.10 INITIAL_LOAD parameter defaults to NO 1–4
1.11 SYNCH_ID no longer required 1–4
1.12 Example Files moved to a subdirectory 1–4
1.13 commonName can have UNIQUE_FIELD processing 1–4

Chapter 2 Problems fixed in V1.4

2.1 BIND_DSA parameter for RFC1006 addresses 2–1
2.2 Modifying the same records in TRANSACTION mode 2–1
2.3 Problems extracting large amounts of X.500 entries 2–1
2.4 Extra characters added to ENCODE_DELIM processing 2–2

Chapter 1
V1.4 New Features

1.1 Limited Use License Support Added

XDSU V1.4 can now support multiple license levels based on the maximum number of X.500 entries that XDSU can import or export. This number is defined in the "Product Token" field of the license PAK. A value in this field determines the maximum number of entries that can be imported or exported. No value in this field means an unlimited number of entries can be imported or exported (as in previous versions of XDSU).

As of this writing, the different levels and prices have not yet been determined, although the unlimited license price and part number has not changed.

1.2 CHANGES Mode Optionally Compares X.500 against Foreign Directory

In export CHANGES mode, the OLD_METAFILE is a copy of what was in X.500 during the prior changes run. This is compared against the current state of X.500, which is extracted into the NEW_METAFILE, and changes export files are generated. For the next XDSU run, the NEW_METAFILE is renamed to the OLD_METAFILE, etc.

However, XDSU has no knowledge that the changes generated will get correctly applied to the foreign directory or database. XDSU would have to compare X.500 against the foreign directory directly to see which changes should be generated.

In XDSU V1.4, therefore, export CHANGES mode can now accept an OPTIONAL foreign directory extract file, which can be compared against X.500 to generate changes the to be exported back to the foreign directory. This could be used for foreign directories which are not reliably guaranteed to properly accept changes or can fall out of synch somehow with X.500 due to external factors. The foreign directory, however, needs to able to generate a full export of entries which were previously imported from XDSU.

To use this option, four new XDSU_CONFIG.DAT parameters for CHANGES mode have been added: RDF_FILE, INPUT_FOR_DIR_FILE, NREC_SUBSCR, and DN_SEQUENCE. These will be used to create the OLD_METAFILE. The OLD_METAFILE and NEW_METAFILE

parameters can be left blank as these will now be temporary files that will not be needed from run to run.

If not using this new option (as before), RDF_FILE must be blank and values for OLD_METAFILE and NEW_METAFILE must be specified.

1.3 Case Changes Optionally Recognized

In previous versions of XDSU, when XDSU imported data into X.500, an attribute was not modified if all that was different was the case of the value. This was consistent with X.500 attributes which are usually case insensitive.

However, because other foreign directories or databases, which XDSU may be synching X.500 with, may be case sensitive, XDSU may now be able to make case changes if desired.

A new, optional XDSU_CONFIG.DAT parameter, RECOGNIZE_CASE_CHANGES, has been added for V1.4. The default will be NO, which will maintain the current functionality of ignoring case changes. If YES, however, in IMPORT and TRANSACTION mode, any case changes will result in the modification of the attribute. And if the distinguished value RDN (i.e. the last RDN value in the DN) changes case, the X.500 entry will be deleted and added with the new case.

In EXPORT and CHANGES mode, setting this parameter to YES will cause case changes to be recognized and exported to foreign directories or databases.

1.4 More Detail Logs for CHANGES mode Added

In V1.4, if VERBOSE=YES in CHANGES mode, XDSU will write to stdout/sys$output the DN of each entry that is used in the DEL, ADD, or MOD files. This is similar to the verbose output used in IMPORT mode.

This allows XDSU to provide easier auditing and reporting of activity during CHANGES mode, which has been requested by some customers.

1.5 Substring filter ability for GET_VALUE added

In V1.4, in EXPORT or CHANGES mode, XDSU now allows specifying a wildcard for the GET_VALUE parameter. For example:

```
GET_FIELD=surname
GET_VALUE=A*
```

This would return all entries where surname begins with 'A'.

```
GET_VALUE=*A*B*
```

This would return all entries where surname contains an 'A' followed by a 'B'

Also, specifying a approximate name match is supported.

```
GET_VALUE=~SMITH
```

This would return all entries where surname is approxiamately SMITH.

This feature has also been implemented for the *_MATCH parameters (e.g. ORGNAME_ MATCH, ORGUNIT_MATCH, etc.) in EXPORT and CHANGES mode. The attributes used for GET_VALUE and the *_MATCH parameters must support substring and approximate match to utilize this feature.

1.6 MHS-OR-ADDRESS Syntax Comparision Restriction Removed

In previous versions of XDSU, if the format for specifying MHS-OR-ADDRESSES SYNTAX Attributes in the input file did not exactly match what was extracted a modification was made during each import - even if the value encoded to the same internal value.

In V1.4, XDSU will now compare the internal values so needless modifies will no longer be performed.

1.7 Script Files Added to make running XDSU easier

In V1.4, for the Digital UNIX version of XDSU only, several optional script (command) files have been added to the XDSU root directory which can be used to make running xdsu instances easier:

```
do_import  - script to do a single xdsu import
do_trans  - script to do a single xdsu transaction mode import
do_export  - script to do a single xdsu export
do_change  - script to do a single xdsu changes mode export
do_change1  - script to do a single xdsu changes mode export
     where X.500 is compared against a foreign directory
do_archive  - script to archive data files (called by others)
do_send_mail  - script to send mail (called by others)
do_synch  - example master synch script, which calls do_import, do_export,  etc.
```

The output from the do_* scripts produces a good overview of the synch process. So this output could be captured as a synchronization summary report. The detaled XDSU output is also captured in the individual directories (xdsu.log) if needed.

See the example do_synch script and the comments in the do_* scripts for more information on how to use the optional procedures.

1.8 New XDSU subdirectories added

In XDSU V1.4, for the Digital UNIX version of XDSU only, three general purpose subdirectories are created during installation which help organize XDSU data.

```
/var/dxd/xdsu/xfer_in - input files (exports from other systems)
/var/dxd/xdsu/xfer_out - output files (imports to other systems)
/var/dxd/xdsu/logs - central logging directory
```

Use of these directories is optional but are helpful if using the do_* script files. The xfer_in directory provides a central place for all foreign directory exports to be placed which are then used as import files into X.500. The xfer_out directory provides a central place for all X.500 exports to be placed which are then used as foreign directory imports. The logs directory provides a central place to collect all log files.

1.9 XDSU_CONFIG_INSTANCE procedure improved

In V1.4, the XDSU_CONFIG_INSTANCE command procedure has been improved to automatically edit the XDSU_CONFIG.DAT file with default values based on the XDSU mode type (IMPORT, TRANSACTION, EXPORT, or CHANGES) and the name of subdirectory. Also, if an IMPORT or TRANSACTION mode instance, the Group ID value will be added to XDSU_VALID_ID.DAT.

1.10 INITIAL_LOAD parameter defaults to NO

During IMPORT or TRANSACTION mode, if the INITIAL_LOAD=YES, then the GroupID is added to XDSU_VALID_ID.DAT. Since the Group ID can now be entered into the XDSU_VALID_ID.DAT using the XDSU_CONFIG_INSTANCE procedure, the INITIAL_LOAD parameter is no longer required and defaults to NO.

1.11 SYNCH_ID no longer required

The SYNCH_ID and SYNCH_ID_FIELD parameters in XDSU_CONFIG.DAT are now optional.

Although it is recommended that all X.500 entries that XDSU imports be tagged with a SYNCH_ID, it isn't really necessary since there is also a group id attribute that is stored with every entry.

1.12 Example Files moved to a subdirectory

In XDSU V1.4, for the Digital UNIX version of XDSU only, the example files are stored in a subdirectory of the XDSU root directory (/var/dxd/xdsu/examples), instead of in the XDSU root directory.

If upgrading from a previous version of XDSU, old example files in the xdsu root directory should be removed.

1.13 commonName can have UNIQUE_FIELD processing

In XDSU V1.4, the Distinguished Value RDN (DVAL) attribute, which is, by default commonName, can be used in UNIQUE_FIELD processing. Prior to V1.4, only attributes which were not RDN values in XDSU_LAYOUT.DAT could be used in UNIQUE_FIELD processing.

Chapter 2
Problems fixed in V1.4

2.1 BIND_DSA parameter for RFC1006 addresses

In V1.4, specifying a RFC1006 presentation address in XDSU_CONFIG.DAT for the bind_dsa parameter will not work correctly. An example of an RFC1006 address is:

```
"DSA"/"DSA"/"DSA"/RFC1006+16.36.56.76,RFC1006
```

There are still some presentation addresses that are not correctly handled by the BIND_DSA parameter (e.g. non-Digital DSA's). These addresses should be specified in the dua defaults file instead and leave the BIND_DSA blank in XDSU_CONFIG.DAT.

2.2 Modifying the same records in TRANSACTION mode

In previous versions of XDSU, in TRANSACTION Mode during the Modify Pass, if the same entry occured more than once in the Modify Input File, only the initial change was definitely made while the subsequent changes may have been ignored.

In V1.4, XDSU will now compare an entry in the modify input file with what is current in X.500 instead of what X.500 was like at the start of the XDSU run.

2.3 Problems extracting large amounts of X.500 entries

In previous versions of XDSU, extracting a large number of entries from X.500 had problems.

The first problem was that if XDSU or the DSA process ran out of resources (usually virtual memory) during the search, XDSU would return with 0 records extracted instead of an error. In V1.4, a fatal error is now generated and the appropriate error message is reported.

The second problem was that XDSU would get an EMS_FETCH error or abort during an extract after extacting 40000+ entries. (Note: the minimum number of entries which caused the problem was variable depending on the number of RDN values in the DN's.)

This problem has been fixed in V1.4, but will require a higher virtual memory quota for the XDSU process.

2.4 Extra characters added to ENCODE_DELIM processing

In addition to \{}¹©, several other 8-bit ascii codes must be encoded when storing values using teletexStringSyntax and ENCODE_DELIM is specified. They are (in hex) 0xA0, 0xA6, 0xAC, 0xAD, 0xAE.