



LTF

*A Language Translation Facility for the
IBM iSeries*

Version 6 Release 4

March, 2007

COPYRIGHT

© Copyright 2007, Navan Limited. All rights reserved.

Limited rights to copy the present work are hereby granted by the copyright owner named above. Accordingly, there is hereby granted the right to make a limited number of additional copies solely for the internal convenience of the recipient; no copies may otherwise be made. In particular, no copies may be made, no derivative works may be created and no compilations of the subject work may be created for purposes of republication, for redistribution, for sale, for rental, for lease or for any profit motivated activity whatsoever including the use of this work in support of or in conjunction with any service or service offering.

™ IBM, eServer, i5, i5/OS, iSeries, OS/400 and DB2/400 are trademarks of International Business Machines.

™ Microsoft, MS and Windows are trademarks of Microsoft Corporation.

Information in this document is subject to change without notice. The latest version of this manual can always be read/downloaded from the Navan Internet Website, which can be found at

<http://www.navan.co.uk>

VERSION 6 RELEASE 4, MODIFICATION 2

This edition applies to Version 6 Release 4, Modification level 2 of the LTF Program Product.

Unless otherwise stated, all pages in this edition are to Version 6, Release 4 and refer to the iSeries Operating System, Version 5.3 or greater.

LTF Version 6 Release 4 is certified to be operationally compliant with i5/OS V5R4.

Any references in this manual to Navan Utilities or NUTIL refer to standard Navan Utility products.

Information on these products can be obtained from your nearest Navan agent.

TABLE OF CONTENTS

LTF - An Introduction	1
LTF Operation	3
But how does it work?	3
Summary of LTF Facilities	6
LTF Installation Procedures.....	7
Installing LTF.....	8
Installation Notes.....	10
Security Auditors note	10
Effect of user modifications on warranty	10
LTF reports.....	10
Object compression.....	10
Object ownership.....	11
Operating System configuration.....	11
LTF internal security checking	11
Defining an LTF user profile	12
LTF Operations.....	13
The STRLTF (Start LTF) command	14
LDFNLNG - Defining the Language to use	15
LDFNSYS - Defining the System to use	17
The LTF main menu	19
Converting a DDS source member to use message referenced literals.....	20
LCVTDDS - the Convert DDS Literals command	21
LTRXLIT - the Translate Literals command	24
Searching for the Usage of a word in a literal	31
Searching for the usage of a word in a Translation	32
Correcting Words in the Translations.....	33
Building/Rebuilding the Translation Environment	37
Creating individual Display and Printer file objects.....	38
Rebuilding the Word Indexes	39
Printing a list of Literals/Translations	39
Moving other 'text' objects into the Translation library	40
Working with converted source	44
Accessing the IBM Programmers Menu	45
Appendix.....	46
Accessing LTF from within the PDM utility	46
General Notes on Translating Applications	47
Use of special characters.....	47
Abbreviations.....	48
Literals in arrays and program variables.....	49
Characters for options and parameters	50
Date Formats.....	51
Date handling routines	51
Currency formats.....	56
Printer files.....	56
Error Messages	57

Help Text, User Guides and Documentation	57
Utility Commands	58
Using the LCRTDDS command to convert an Internally Described Printer File.....	58
Translating Literals for a specific source member	63
Business Contingency Planning	64
Support Contacts	65

LTF - An Introduction

The primary objective of LTF is to provide programmer-level tools to assist in converting iSeries user application displays and reports from being shown in one user language:

UPD_MST2	Item Master File Maintenance	
Item number	50246-E	Status.....: L
Description	FILTER	
Date introduced ..	1/01/88	Current cost
New model?	NO	Average cost
Country of Origin:	DE	Retail Price.....: 22,87
Product group	00685	
Duty code	8421230	
Reference	Join	
Nato number.....:		Part Discount Code: G
Vehicle code		Sales tax code ...: 1
Vehicle set		Price freeze flag..:
Lead time (days)..:		Sell Pack Size....: 10
Inventory UOM:		Surcharge Amount..:
Purchase UOM		Reason...:
Weight per unit ..:	0,482	Prod. start
In-line Part No...:		Prod. end
Service Exch Code.:		Campaign code
Discontinued Part?:		Season code
Obsolete Part Code:		Order at request..:
Price Sim Code....:		Default Supplier...: 29980
Enter - to change	F3=Exit	F12=Cancel to prompt

In to being capable of translation into one *or more* other user languages:

UPD_MST2		Vedlikehold - Varefil	
Varenummer	50246-E	Varestatus.....	L
Varenavn.....	FILTER		
Dato innfart.....	1/01/88	Gjeldende kostpris:	
Ny modell?	NO	Gj.snitt. kostpris:	22,87
Opprinnelsesland..	DE	Prisliste 1.....	41,00
Produktgruppe	00685		
Tollkode.....	8421230		
Referanse	Join		
Nato nummer.....		Varerabattkode....	G
Kjoretoyskode.....		M V A - kode.....	1
Kjoretoysett.....		Prisstoptegn	
Ledetid (dager)...		Kvantumsant. salg :	10
Enhet - salg.....		Tilleggsbelop.....	
Enhet - innkjop...		Arsak.....	
Vekt pr. enhet....	0,482	Produksjon-start..	0/02
Ensartet Varenr...		Produksjon-slutt..	0/02
Byttekode.....		Kampanjekode.....	
Utgatt ?.....		Sesongkode.....	
Kuranskode.....		Best. ved foresp..	
Prissimul.kode....		Hovedleverandor...	29980
Entre - for endrin		F3=Retur til meny	F12=Retur til prompt

using as little programmer development effort as possible.

Most software today is designed and developed with only one language in mind - that of the end user of the application. It is normally only considered when developing a new software application if that application is intended to be distributed to other countries, or if users of the system are required to work in more than one language.

Provided that the designers of a new application take the multi-lingual requirement into consideration when in the design stage, the facilities that exist on the IBM iSeries computer can more than adequately cope with that requirement; the supplied programming development tools provide the necessary facilities to cover all phases of the development. Some of the facilities may not be the most user-friendly, but they do work.

But what if a requirement exists for the conversion of an existing application into a foreign language? Can it be done?

The answer is, of course, yes - but at a cost. Basically the concept of conversion is straightforward enough; it is simply a matter of converting any screen or report based text - Literals - from the local (or base) language into the foreign (or target) language by some means or another. The easiest means of doing this is for a programmer to sit down and change the OS/400 display file and printer file source statements. However, this is neither the cheapest nor the most effective means of doing the translation. Firstly, you would have to find a multi-lingual programmer willing to sit down and do such a 'menial' task. Secondly, the method does not allow for system modifications or upgrades, which would require two sets of source to be maintained in parallel. Finally, it makes the task of translation into a third, fourth or fifth language no simpler.

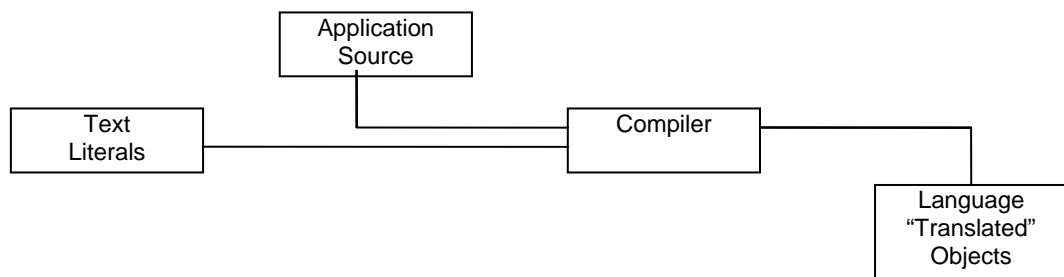
The purpose of LTF (Navan Language Translation Facility) is to provide a utility to assist in this translation process. It is designed specifically for the OS/400 environment and takes full advantage of the facilities provided by the iSeries.

LTF allows an *unlimited* number of systems or applications to be translated and into an *unlimited* number of user languages, completely independent of one another.

LTF Operation

But how does it work?

When the iSeries was designed, it was always intended to be a multi-lingual machine. The method IBM developed to achieve this was to ensure that all Literals were capable of being externally defined from any object source and then merged into an object as it was being created. Thus, once the application was completed, all that had to be done was to load the externally defined literals into the final system objects:



The means of externalising literals is via the Message File object. Any literal that would appear to a user via a 'compiled' object can access a message file to retrieve these literals:

- A command can use a prompt message ID
- A display or printer file can use the MSGCON keyword
- A display file can use the MSGID keyword

When MSGCON is used, what happens at compile time is that the message is retrieved and the first level text from the message is used to provide the literal text to display.

This method is known as **Early Binding**; the literal is bound into the object at compile-time.

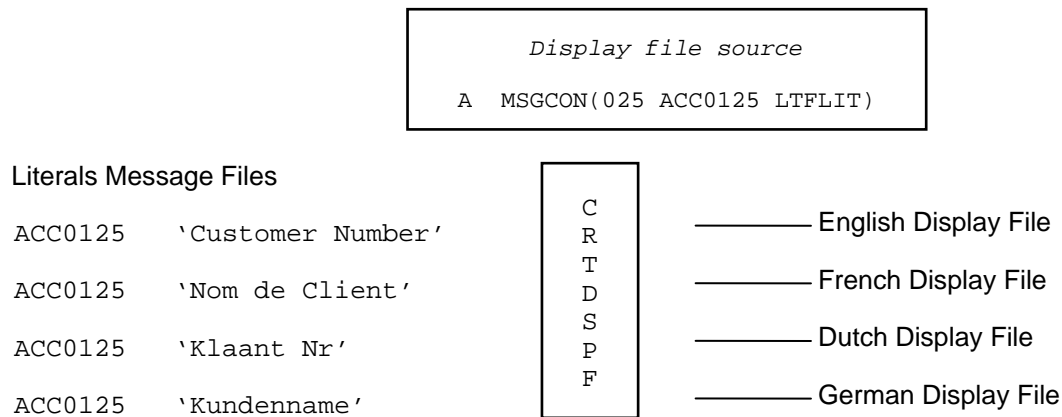
'Normal' DDS Source:

A				6	7'Customer Number'
A		MCCUS	R	B 6	30REFFLD(CUST_NO)
A	41				ERRMSGID(ERR0023 ACMSGF)
A	42				ERRMSGID(ERR0024 ACMSGF)
A	43				ERRMSGID(ERR0005 ACMSGF)
A	44				ERRMSGID(ERR0265 ACMSGF)
A					DSPATR(PC)

'Externalized' DDS Source:

A				6	7MSGCON(025 ACC0125 LTFLIT)
A		MCCUS	R	B 6	30REFFLD(CUST_NO)
A	41				ERRMSGID(ERR0023 ACMSGF)
A	42				ERRMSGID(ERR0024 ACMSGF)
A	43				ERRMSGID(ERR0005 ACMSGF)
A	44				ERRMSGID(ERR0265 ACMSGF)
A					DSPATR(PC)

If you have one message file in one library with your English literals, another message file in another library with your French literals, and so on, then it is just a matter of correctly adjusting the library list prior to compiling an object in order to get one language or another:



This is the method IBM developed for providing the iSeries with a multi-lingual capability; this is also the method that LTF uses to give you the same facility.

A major advantage of using this methodology is that you can always be sure that IBM will support it. If IBM use it themselves they have added incentive to keep it up to date (and possibly to improve it). For example, when you use the MSGCON keyword in display file source the OS/400 Screen Design Aid will retrieve the literals from the message file when it displays the record formats.

There is another DDS keyword available, which can also be used for literal translation purposes - MSGID.

The difference between MSGCON and MSGID is that MSGID is **Late Binding**, in other words the literals are retrieved from the message file when the screen is displayed to the user. MSGCON is used by default in LTF for technical reasons; mainly that there is total compatibility between the original display file object and the 'converted' display file object. The use of MSGID requires a recompile of the associated program (and thus stops you from using the original display file object). Also, MSGID cannot be used for printer files.

MSGID can, however, be requested within LTF should you wish to use this with your Display files.

There are other means of achieving the same finished product. Another example of a **Late Binding** facility (literals are loaded into the object when it is used) is to use a Physical file to contain all Literals and the program then loads the literals into screen variables when the program is called.

The advantage of a late binding method is that no compile phase is required - when you change the literal in the file the new translation is available immediately. However the disadvantage is performance - it takes time to retrieve the literals.

One final note: LTF works on DDS source. If you do not have access to the source members for your application you cannot process the application within LTF.

Summary of LTF Facilities

LTF is designed to assist you in translating an application for use by an end-user of the application.

The LTF utility consists of five phases:

- Ensuring all DDS source is externally defined
- Converting DDS source to 'externalise' literals

- Providing a Literal Translation facility

- Providing a 'mass re-create' facility
- Allowing facilities to translate other text objects

All options are available via a normal IBM Command Entry screen, but for ease of use in a translation environment the main functions of LTF have been grouped together in the form of a menu. To access this menu from any command entry point, just type in LTF/STRLTF and press the F4 key to prompt the command, as follows:

```
ADDLIBLE LIB(LTF) POSITION(*LAST)      ...ENTER
LTF/STRLTF                            ...F4 (prompt)
```

When using the MSGCON DDS keyword in the converted DDS, the Display/Printer file object that is created after the translation process is directly compatible with the original (untranslated) object and thus no recompile of related programs is necessary.

To switch between languages is simply a matter of switching the user's library list.

LTF Installation Procedures

Please read this section of the document thoroughly before commencing the installation of LTF.

All steps in this installation guide must be performed using the System Security Officer user profile, or by using a user profile assigned *ALLOBJ special authority.

Certain programs in LTF adopt authority when in use. This authority adoption must be retained if the system is to operate as intended.

Installing LTF

The tape/savefiles supplied contain a complete (replacement) LTF library and can be installed automatically. If you already have an old version of LTF on your iSeries, your data will be loaded from the old version into your new version, by the installation command. Installation is performed as follows:

1. If already installed, ensure that no users are using any LTF functions. No activity is allowed whilst this installation is being performed.
2. Sign on as the System Security Officer QSECOFR, or as a user profile that has *ALLOBJ special authority. This installation MUST be performed by a user with *ALLOBJ authority for all programs to function as designed.
3. If LTF is in your library list, remove it:

```
RMVLIBLE LTF
```

4. Save a copy of your existing version of the LTF library (if you have it already installed):

```
SAVLIB LIB(LTF) ...
```

You are now ready to commence the installation. The installation routine can be run using either a CD-ROM disk, tape, or from savefile objects:

**** IF YOU ARE INSTALLING FROM CD-ROM DISK****

- 5(CD). Load the supplied CD-ROM disk into the iSeries CD drive and install the new version of the LTF library using the LODRUN command:

```
LODRUN DEV(*OPT)
```

The supplied CD can also be used to extract all LTF documentation. Documentation is stored in the \DOCS directory of the CD and can be accessed using the CD-ROM drive of your PC.

**** IF YOU ARE INSTALLING FROM TAPE ****

- 5(tape). Load the supplied tape onto the tape drive and install the new version of the LTF library using the LODRUN command:

```
LODRUN DEV(TAP01)
```

If you are using a tape device other than TAP01, alter the DEvice parameter in the above command accordingly.

**** IF YOU ARE INSTALLING FROM SAVEFILES ****

5(savf). You should have a library called NAVANSAV on your system, which contains the following objects:

QINSLTF	The installation program
LTFSV	Save file containing the LTF library objects
LTDTAARASV	Save file containing the LTF data area

Install the new version of the LTF library using the installation program:

```
CALL PGM(NAVANSAV/QINSLTF) PARM(*SAVF)
```

A pre-check will be made to ensure the objects specified above exist. If they do not, an error message will be sent and the program will end.

6. The procedure will then install LTF. At the completion of the procedure you should receive the message

'LTF installation procedure completed successfully'

If you do not get this message, the installation procedure was processed incorrectly. You can determine the cause of the error by reviewing the low level messages for the job. If you cannot resolve the problem, you should contact your LTF support representative. You will not be able to continue the installation of this version of LTF and you should reinstall your old version of the LTF library (if you have them).

Installation Notes

Security Auditors note

The LTF installation procedure is performed by the QINSTAPP installation program, which can be found at SEQNBR 1 on the supplied tape. Should you need to review the installation procedure for security purposes you can do this as follows:

```
CRTSRCPF  FILE(QTEMP/QCLSRC)

RSTOBJ    OBJ(QINSTAPP) DEV(TAP01) SAVLIB(QTEMP) +
          OBJTYPE(*PGM) RSTLIB(QTEMP)

RTVCLSRC  PGM(QTEMP/QINSTAPP) SRCFILE(QTEMP/QCLSRC)
```

The source for the installation program is then in file QTEMP/QCLSRC, member QINSTAPP. Should you have any security objections to this installation procedure, please advise Navan directly.

It is possible (but not recommended) to install LTF using a modified version of the retrieved CL source but in doing so *you remove any liability from Navan* for the successful functioning of the installation procedure as well as the successful functioning of the LTF program product.

Effect of user modifications on warranty

Any warranty implied or expressed by Navan relating to this product does not apply to any portion of the product altered by someone other than an authorised Navan employee.

In other words, user modifications to the LTF product library are not covered by any Navan warranty or support agreement.

LTF reports

As supplied, all LTF printed output will be to a page size of 66 lines, with page overflow occurring at line 60. Should you wish to alter this to conform to a different standard you should use IBM supplied CHGPRTF command, specifying your own definitions as required.

Object compression

The LTF library objects are supplied in compressed form. You should give consideration to decompressing the objects in order to improve user access times. Refer to the CL Reference manual for a discussion of the Decompress Objects (DCPOBJ) command.

Object ownership

Unless there are special processing considerations, objects in LTF are owned by QPGMR and are freely accessible to all users on the system (all users have free public access). Where a command performs an operation that is not normally allowable by a 'low' authority user, the program will adopt sufficient authority in order to ensure the function works correctly. All LTF command objects are set to ALWLMTUSR(*NO), so that limited capability users can not request them from the command line on a menu.

Operating System configuration

LTF is designed to work with a standard iSeries operating system ('OS/400'), as supplied by IBM. Should any local changes have been applied to any portion of this standard operating system, there is no guarantee of correct operation of the LTF system.

Such local changes to OS/400 include any alterations to system object authorisations; changing command parameter defaults; changing command processing environment defaults.

LTF internal security checking

Certain functions within LTF are 'locked' to the licensing information provided with your copy of LTF. Internal system checks within LTF functions are made against the following data:

- CPU Serial Number
- CPU Model Number
- System Date

This is to ensure your copy is an authorised version of LTF. You should be aware that if any of the CPU control information on your iSeries changes, or you attempt to use the version licensed to your CPU on another iSeries, functions within LTF may cease to work.

Your licensing information is displayable by typing CALL LT9000 in the command line of the LTF main menu.

Should you be upgrading your existing CPU to another model, or moving on to another iSeries completely, you will need to contact your Navan representative for an upgrade authorisation, which may be subject to additional licensing fees.

Demonstration versions of LTF will automatically cease to function when the expiry date has been reached.

By installing a demonstration version of LTF, you are accepting that Navan and its representatives take no responsibility for data lost within LTF if your demonstration copy of LTF reaches its expiry date, or if an internal security violation is detected by LTF.

Defining an LTF user profile

It is recommended that you create a unique user profile for each user planning to use LTF. Each system (user application) you intend to process should have its own user profile *template* (or definition), to ensure the library list is correct for that system.

The best way of controlling library list definitions for user signon is via a job description, so for each system you should create a job description in LTF. The library list defined in that job description should be the normal development library, but with LTF added before QTEMP.

Once you have the required 'LTF development' job description(s) defined you can then create the user profile(s). The basic parameters for each user profile are as follows:

```
CRTUSRPRF  USRPRF(ltf_user_ID)
              INLPGM(LTF/INLPGM)
              INLMNU(*SIGNOFF)
              CURLIB(*CRTDFT)
              JOBD(LTF/ltf_development_job_description)
              GRPPRF(development_group_profile)
              OWNER(*GRPPRF)
```

Using this method the user is automatically assigned the correct library list; is logged straight in to the LTF main menu and has correct object authorities.

LTF Operations

All options are available via a normal IBM Command Entry screen, but for ease of use in a translation environment the main functions of LTF have been grouped together in the form of a menu.

To access this menu from any command entry point, just type in LTF/STRLTF and press the F4 key to prompt the command, as follows:

```
ADDLIBLE LIB(LTF) POSITION(*LAST)    ...ENTER
LTF/STRLTF                          ...F4 (prompt)
```

The STRLTF (Start LTF) command

```

                                Start LTF (STRLTF)

Type choices, press Enter.

System to Translate . . . . . ACCTS
System Option . . . . . *CREATE

Translate to Language . . . . . 2933
Language Option . . . . . *CREATE

```

When a user normally accesses the LTF facility via this command it would be sufficient to allow the command parameter defaults, as this would access the current language and system in use (*PRV).

However the first time the LTF utility is used, or the first time a new user signs on to LTF, the prompt requires specific entries to be made:

System to translate	This is an identifier to LTF that uniquely defines this application.
	*PRV - Use the system that was in use the last time this user accessed LTF
	*SELECT - Display a list of systems that have been defined to LTF and allow the user to choose one.
System option	Valid entries are:
	*CREATE - Create the new system
	*USE - Use the previously defined system.
Language to translate	This is an identifier to LTF that uniquely defines this language.
	*PRV - Use the language that was in use the last time this user accessed LTF
	*SELECT - Display a list of languages that have been defined to LTF and allows the user to choose one.
Language option	Valid entries are:
	*CREATE - Create the new language
	*USE - Use the previously defined language.

(In the above example, language 2933 will be created and system ACCTS will also be created. Later prompts will then ask you to define these further.)

LDFNLNG - Defining the Language to use

```

                                Define language in use (LDFNLNG)

Type choices, press Enter.

Translate to Language . . . . . > 2933
Language Option . . . . . > *CREATE

Language Title . . . . . Norwegian Translations

```

This command is used to define a new language to LTF. It is automatically prompted by the STRLTF (Start LTF) command if you specified language option (*CREATE).

The command parameters are:

Language to translate	This is an identifier to LTF that uniquely defines this language. *PRV - Use the language that was in use the last time this user accessed LTF *SELECT - Display a list of languages that have been defined to LTF and allows the user to choose one.
Language option	Valid entries are: *CREATE - Create the new language *USE - Use the previously defined language.

For *OPTION(*CREATE)*:

Language title	Enter a language title (description). This title will appear on any language select screens
----------------	---

When creating a new language, LTF will create a new library on your iSeries named LTFaaaaaaa, where aaaaaaa is the language identifier you specified in the command.

The library will initially contain the file objects necessary to hold translations for that language. As screens and reports also get translated the objects will also reside in this library. This is the library that will be in the user's library list.

Language Library Naming Conventions

You may use any identifier you wish when creating a new language library providing, of course, that the resulting library name generated is a valid name under OS/400 rules. IBM have already defined a set of National Language codes that you may wish to follow:

Code	Language
2911	Slovenian
2912	Croatian
2922	Portuguese
2923	Dutch Netherlands
2924	English
2925	Finnish
2926	Danish
2928	French
2929	German
2931	Spanish
2932	Italian
2933	Norwegian
2937	Swedish
2954	Arabic
2956	Turkish
2957	Greek
2958	Icelandic
2959	Greek
2961	Hebrew
2963	Belgian Dutch
2966	Belgian French
2972	Thai
2975	Czech
2976	Hungarian
2978	Polish
2979	Russian
2980	Brazilian Portuguese
2981	Canadian French
2994	Slovakian
2998	Farsi

Thus when you create a language library for your Norwegian translations, using this numbering system would cause a library named LTF2933 to be created. This library will contain the objects necessary for an end user to operate the user application in Norwegian.

LDFNSYS - Defining the System to use

```

                                Define system in use (LDFNSYS)

Type choices, press Enter.

System to Translate . . . . . > ACCTS
Option . . . . . > *CREATE

System Description . . . . . Accounts System
Abbreviated system ID . . . . . > ACC
Installation type . . . . . *LOCAL

```

This command is used to define a new system to LTF. It is automatically prompted by the STRLTF (Start LTF) command if you specified system option (*CREATE). Command parameters are:

System to translate	This is an identifier to LTF that uniquely defines this application.
	*PRV - Use the system that was in use the last time this user accessed LTF
	*SELECT - Display a list of systems that have been defined to LTF and allows the user to choose one.
System option	Valid entries are:
	*CREATE - Create the new system
	*USE - Use the previously defined system.

For OPTION(*CREATE):

System description	Enter a system description. This description will appear on any system select screens.
Abbreviated system ID	Enter a system abbreviation. This code must be unique for this system. The abbreviation is used to prefix all literal identifiers generated for this system.
Installation type	Enter an Installation type, which can be either:
	*LOCAL - This is a local copy of a system which is normally maintained elsewhere, or
	*MASTER - This is the master copy of a system.
	The different installation types have their own unique set of Literal ID codes, to ensure that local modifications to a system do not conflict with the master set when a new release of the system is installed at a local site.

When creating a new system, LTF will create a source file in library LTF to contain any converted DDS source members. This file will be named QDDSaaaaaa, where aaaaaa is the system ID you specified in the command. The file will initially have no members. Members will be loaded in this file via the LCVTDDS (Convert DDS) command.

System naming conventions

You may use any identifier you wish when creating a new system identifier and system abbreviation providing, of course, that the resulting source file name generated is a valid name under OS/400 rules.

Literal numbering conventions

Literals within LTF are stored and accessed by a unique, 7 character, 'Literal ID' identifying code. This code is made up of the abbreviated system ID you specify on the LDFNSYS (define system) command followed by a sequential number, starting at 1.

The length of the Abbreviated System ID you specify will determine the amount of literals you have available to a specific system:

For a 3 character abbreviation the Literal ID code will be *aaannnn*

where *aaa* is the abbreviated system ID
nnnn is the literal number

For a 2 character abbreviation the Literal ID code will be *aannnnn*

where *aa* is the abbreviated system ID
nnnnn is the literal number

For a 1 character abbreviation the Literal ID code will be *annnnnn*

where *a* is the abbreviated system ID
nnnnnn is the literal number

Literal numbering ranges

It is important to consider the size of your application when deciding on the length of the abbreviated system ID you specify. While 3 characters provides a 'meaningful' prefix to the Literal ID code, it also restricts the amount of literals that can exist within the system.

Based on the length of the abbreviated system ID, the available numbering ranges for literals within a system defined in LTF will be:

	*MASTER set	*LOCAL set
3 character abbreviation	0001 - 7999	8000 - 9999
2 character abbreviation	00001 - 79999	80000 - 99999
1 character abbreviation	000001 - 799999	800000 - 999999

It is recommended that you restrict yourself to using either a two, or even a one, character abbreviation to ensure the message numbering range can cope with growth in your application.

The LTF main menu

You will then be presented with the LTF Main Menu:

```

LTF                               Language Translation Facility

Select one of the following:

Main options
  1. Convert source member to use message literals           LCVTDDS
  2. Translate the literals                                 LTRXLIT
  3. Rebuild the translation environment                    LRBLTRXENV
  4. Create display/printer files from converted source     NCRTOBJ
  5. Create display/printer files based on literal usage    NXRFCRTOBJ
  6. Rebuild the word cross reference                       LRB010
  7. List all literals/translations                         LPRTLIT

Support functions
  10. Create a duplicate object in the translation library   CRTDUPOBJ
  15. Modify a message file (error messages, etc)          WRKMSGD
  16. Modify a data file (parameters, etc)                 UPDDTA
  17. Work with converted source                           WRKMBRPDM
  20. Use the programmers menu                             STRPGMMNU
  35. Display LTF installation attributes

Other options
  90. Sign off

Selection or command
====>

F3=Exit   F4=Prompt   F6=DSPMSG   F9=Retrieve   F12=Cancel   F13=Display Attributes
F14=WRKSBJOB  F15=Statistics  F18=WRKSPLF  F20=Set System  F21=Set Language
© Copyright Navan Limited, 2006

```

The LTF Main Menu is the main access point for LTF activities.

When you take any of the options shown you will be working in the system and language specified on the STRLTF command. You can change to another language or system by taking the appropriate function key.

From this screen you can also access other functions normally associated with a programming environment. The menu operates in a similar way to an IBM 'GO' menu (although it *must* be accessed via the STRLTF command).

Converting a DDS source member to use message referenced literals

The LCVTDDS (Convert DDS) command accesses an existing DDS source member (Printer File or Display File) and uses it to generate a new source member in which any literal strings are changed to MSGCON or MSGID statements.

The original source member is not altered in any way by LTF; instead it is used as a template from which to create a new (revised) source member.

For every literal encountered in the DDS source, the command generates a message reference and changes the 'literal constant' text source line into a MSGCON or MSGID source line (dependent on the parameter you enter in the LCVTDDS command prompt):

Original source line:

```
A                15    32'Part Number'
```

converts to:

```
**                vv...vv'Part Number'
A                15    32MSGCON(011 ACC1051 LTFLIT)
```

or (using MSGID):

```
**                vv...vv'Part Number'
A    LIT001  11A  O    15    32MSGID(ACC1051 LTFLIT)
```

Existing literals are 'optimized' automatically; that is, if the same character string was previously used to generate a message then that previously generated message will be referred to (rather than generating another 'duplicate' literal).

Accordingly the translation process becomes simpler, as it is only necessary for a translator to enter a translation for a literal once.

LCVTDDS - the Convert DDS Literals command

LTF menu Option 1 processes the Convert DDS Literals (LCVTDDS) command.

```

                                Convert DDS Literals (LCVTDDS)

Type choices, press Enter.

From source file . . . . .> QDDSSRC
  Library name . . . . .> ACCTSSRC
Member name . . . . .> ACC01
To source file . . . . .> QDDSSACCTS
  Library name . . . . .> LTF
Allow literal sharing? . . . . .> *NO
DDS Keyword type . . . . .> *MSGCON
Rebuild keyword index? . . . . .> *YES

```

The Convert DDS literals command accesses an existing DDS source member (Printer File or Display File) and uses it to generate a new source member in which any literal strings are changed to MSGCON or MSGID statements. For every literal encountered the command generates a message reference and changes the literal text source line into a MSGCON or MSGID source line (dependent on the keyword type you specify to be used).

Command parameters are as follows:

From source file	This is name of the source file that contains the original DDS source member.
Member name	This is the name of the DDS source member.
To source file	This is the name of the source file that LCVTTDDS will load the converted DDS source member into. It cannot be the same file used in the 'From Source File' parameter.

- Allow literal sharing? The Allow Literal Sharing parameter defines whether the conversion should allow the referencing of equivalent literals in another system defined to LTF.
- An entry of *NO means that the literals in this system are unique. Equivalent literals in other systems should not be referenced. An entry of *YES allows the use of equivalent literals defined in other systems.
- DDS keyword type This parameter determines which of the two DDS keywords is to be used, MSGCON or MSGID.
- The default is to use the MSGCON keyword.
- Note that *MSGID cannot be used for printer files. If you request *MSGID and a printer file source member is to be processed, LTF will switch to *MSGCON and advise you (so that if you are processing a source file generically you can still use *MSGID if you wish. This way all display file source members will be processed as *MSGID and any printer file source members will be processed as *MSGCON.
- Rebuild keyword index? LTF maintains a cross-reference file that can be used in the Literals Translation phase. Inquiry functions are available to show every occurrence of a word or a phrase, to assist the translator should there be questions about meaning or context.
- To ensure these are kept up-to-date this parameter should be left as the default (*YES). However, if you do not want to use these Inquiry facilities you can set the parameter to *NO.

The LCVTDDS command can not be processed interactively. If it is requested from a command entry line, it will automatically submit itself for processing after the user has pressed enter.

The converted member generated in the 'To source file' will have the same member name and member type as the original member, and the member text will also be the same. A comment is inserted as the first record of the converted source member, which gives details relating to the source conversion processing.

The 'DDS keyword to use' parameter will determine how the converted source is to be created. The actual processing performed is functionally the same, but the resulting DDS will work differently:

- If you use KWDTYP(*MSGCON), the MSGCON keyword will be used within the converted DDS. When the display/printer file is created (using the CRT... command) the text to be used is retrieved and 'compiled' into the created object (**early binding** technique). This newly created object is compatible with the original display/printer file object, and thus no re-creation of any associated programs would be required.
- If you use KWDTYP(*MSGID), the MSGID keyword will be used within the converted DDS. The text is not accessed when the display file is created (using the CRTDSPF command), it is retrieved when the display is actually displayed on a users workstation (**late binding** technique). The newly created display file is not compatible with the original, therefore any associated programs must be recompiled to avoid level-check errors.

Limitations and notes

Literals within LTF can be a maximum length of 132 characters. Should the LCVTDDS command detect a literal longer than this it will process the literal as if it was 132 characters long and insert a comment line in the converted source member. This comment line is syntactically incorrect and will cause a severity 30 error should an object creation be attempted from the source. You must manually alter the original source member to ensure no literal is greater than 132 characters in length.

Should the LCVTDDS command detect a literal longer than 80 characters in a display file source, it will insert a 'warning' comment line in the converted source member. The comment has no affect on the generated source and only serves to warn you that the literal may wrap across multiple lines on the display panel.

An ampersand character ('&') followed by a number, or followed by the letters B, P or N, is not allowed within a literal. This is a special OS/400 messaging 'control value' that has a specific meaning within message text on the iSeries and can not be used within translation literals. Any source member containing this special value will be converted and a comment line will be added to the converted source member. This comment line is syntactically incorrect and will cause a severity 30 error should an object creation be attempted from the source. You must manually alter the original source member to ensure no literal contains a special message control value.

Also, any literal translation you later supply to LTF will not be allowed to contain any of the special messaging control values.

It is recommended that no source changes be performed on the converted source member that was created by the LCVTDDS command.

The preferred way of performing maintenance is to correct the *original* source member and then reprocess it and reload it into LTF using the LCVTDDS command. In this way you never need to worry about maintaining compatibility with the original source member - you just re-process the original source member back through the LCVTDDS command again after any maintenance.

LTRXLIT - the Translate Literals command

LTF menu Option 2 processes the Translate Literals (LTRXLIT) command:

```

                                Translate Literals (LTRXLIT)

Type choices, press Enter.

Display Sequence . . . . . > *LITID
System to Translate . . . . . *CURRENT
Start from Literal . . . . . *START
Source Member Name . . . . . *ALL

```

Command parameters are as follows:

Display sequence	<p>This parameter determines the order in which the literals will be displayed on the list screen. The normal sequence is *LITID (Literal ID) sequence. The advantage of this sequence is that related literals (such as column headings) will be shown one after the other. Other sequences available are:</p> <p>*ALPHA (Literals are displayed in alphabetic sequence); *LITWRD (Literals can be accessed by any word in the literal); *TRXWRD (Literals can be accessed by any word in the translated literal); or *WRDCOR (Allows corrections to be made of a specific word in all literals that use the word).</p>
System to translate	<p>The default of *CURRENT will allow you to translate literals within the system you are currently accessing.</p> <p>An entry of *ALL will allow you to process all literals in all systems. Alternatively, you may enter a specific system identifier to access any other system.</p>
Start from literal	<p>This option allows you to go to a specific point in the literals file.</p> <p>*START will commence displaying literals from the beginning of the file. *LAST will start the display from the last literal that you accessed (this is only valid for sequence *LITID or *ALPHA). Or you can enter a specific literal number (valid for sequence *LITID only).</p>
Source member name	<p>This feature allows you to select and process only those literals that are used in a specific display or printer file. Valid for *LITID sequence only.</p>

Literal Translation, Literal ID sequence

The actual layout of the translation work display will depend upon the display sequence you requested, but all four function in basically the same manner. Below is an example of the *LITID sequence work display:

```

LTX003D1                      Translate Literals
Select   Language: 2929      German Translations System: INVT

Type options, press Enter.      Member . . . . . *ALL      F4 for list
   2=Change   5=Display

Opt Literal Sts   Text Description

   INV1201 ALL

   INV1201      A - Add new part number
   INV1202      C - Change existing part number
   INV1203      D - Delete existing part number
   INV1204      R - Reinstate deleted part number
   INV1205      T - Maintain Descriptive Text
   INV1206      - to proceed
   INV1207      - Return to menu
   INV1208 IGN  STFA5002
   INV1209      Part number .....:
   INV1210      Part Status.....:
   INV1211      Description .....:
   INV1212      Date introduced ..:

                                          More...

F3=Exit   F4=Prompt   F10=Continuous Update   F11=Change View   F13=Switch
F15=Statistics   F16=Search Lit   F17=Search Trx   F20=Initialise Literals

```

The 'list' area of the screen will show 12 literals at a time. You can scroll through the literals by pressing the PAGE UP or PAGE DOWN keys. You can reposition the list to a specific section of the file by entering a literal ID in the 'Position Literal Start' prompt and pressing the Enter key.

The information displayed in the list section is as follows:

Select literal to change (Opt)	If you enter a '5' beside any literal you will be shown the current translation for the literal. If you enter a '2' beside any literal, you will select the literal for processing (see next section 'Entering the Translation').
Literal ID	This is the unique identifier for the literal.
Translation status (Sts)	This is the status of the literal in regards to its translation. The entry in this column may be one of the following: '***' Not yet translated 'IGN' Translation ignored 'RPL' Literal is being replaced ' ' Translation entered
Untranslated text	The first 60 characters of the literal text. An arrow delimiter '<' at the beginning of the literal signifies that the literal has leading blanks and the literal has been 'shuffled up' so that the first non-blank character in the literal is displayed. This function is purely for display purposes. The actual literal remains unchanged.

As previously stated, you can enter '2' beside any literal in the list to select it for processing. If you enter a '2' beside more than one literal you can process multiple literals without having to return back to this screen.

The function keys available on this display are as follows:

- F3 Exit from the Literal Translation command.
- F4 With the cursor positioned on the *Member* name field, this will show you a list of DDS source members defined within this system. Selection of a specific member will only display the literals related to that specific member.
- F10 Continuous Update mode. When you press this function key the command will take you through the updating of every untranslated literal, commencing from the last literal you translated (you can exit the function at any time by pressing F3).
- F11 This display has two views:
View 1 will show 12 literals on the work panel
View 2 will show only 6 literals, but it will also show the current translation for that literal on the line below it
You can toggle between the views by pressing F11.
- F13 Allows you to switch to *ALPHA mode
- F15 Displays a screen showing total translation counts for this system and language
- F16 Search Literals. This command takes you to an Inquiry display that lets you search for occurrences of a literal, right down to the display of the actual DDS source code if you require.
- F17 Search Translations. This command takes you to an Inquiry display that lets you search for occurrences of a translation, right down to the display of the actual DDS source code if you require.
- F20 Sets the current translation to be the same as the base literal, for *any literals* not yet translated.

Entering the Translation

The next screen will be displayed for every Literal that you selected for processing (or for all untranslated literals if you are in continuous update mode):

```

LTX003D2                Translate Literals
Change  Language: 2929   German Translations System: INVT

Literal ID.....: INV1209

Literal Length.....: 19
Number of uses.....:

Original Literal Text:
  Part number .....:

Translation:
  Teile Nummer.....:

Replaced by Lit ID.:

F3=Exit  F9=Ignore translation (no translation required)  F10=Reset
F12=Previous  F13=Switch to *ALPHA  F15=Display Literal Usage

```

This screen shows you the original literal text and provides you with a prompt to enter your translation.

The information displayed in the list section is as follows:

Literal ID	The unique identifier for this literal
Literal length	This is the actual length of the literal. Your translation can not exceed this length.
Number of uses	This information tells you how many DDS source members use this literal.
Original literal text	This is the actual text of the literal, as it was extracted via the LCVTDDS (Convert DDS) command from the original DDS source.
Translation	This is the current translation loaded for this literal. If the text is <i>exactly the same</i> as the original literal it means that no translation has yet been entered, and the original literal text has been loaded into the field as a 'template' for you to work with.
Replaced by literal ID	This feature allows you to replace one literal with another throughout the system. An explanation of this feature is provided below.

The function keys available on this display are as follows:

F3	Exit from the Literal Translation command.
F9	Ignore translation (no translation required)
F10	Reset the translation status, remove current translation.
F12	Return to the previous screen.
F13	Switch to *ALPHA sequence list display.
F15	Display literal usage. See next section of this manual

Notes

F9= Ignore translation:

There are many times when a literal will not need any translation entered; for example the literal 'MO154' has no translatable meaning; it could be a screen identifier code for instance.

When you have a literal that does not require any translation, press F9 on this screen and the translation status will be set to 'IGN' Translation ignored. You can always re-access the literal and enter a translation later if you made a mistake in ignoring a specific literal. The entry of translation text (or the use of the F10= Reset function key) will reset the status of the literal.

F13= Switch display sequence:

All screen sequences allow you to switch to another sequence without having to terminate the LTRXLIT command. If you requested *LITID sequence you can switch to *ALPHA sequence and vice versa, by the use of the F13 function key. If you requested *LITWRD sequence you can switch to *TRXWRD sequence and vice versa, by the use of the F13 function key.

Literal Replacement

This feature gives you the opportunity to group similar literals into using the one translation. For example, take the following literals:

Literal ID	Literal	Length
INV1052	'Part Number'	11
INV2146	'Part No'	7
INV2731	'Part Nr'	7
INV3989	'Article No'	10

For the purposes of this example, all four literals shown have the same meaning. You may enter a translation for each of the four or, alternatively you can group the literals. In this case you can enter SPA2146 as the replacement literal for the other three. The two rules in using this facility are that:

1. The replacement literal length cannot exceed the length of the literal it is replacing.
2. The replacement literal cannot be flagged for replacement itself.

When you flag a literal for replacement by entering a replacement literal ID, the command memorises the fact that you have requested at least one literal to be replaced. When you exit the LTRXLIT command you will be prompted to allow the submission of a batch job which will do the following:

1. Update the translation control files
2. Replace the literal ID in every source member that uses it, with the replacement literal ID.

If you re-process the source member through the LCVTDDS command again you will remove the replacement and the original literal reference will be restored.

Displaying the usage of the Literal

If you use the *F15=Display Literal Usage* function key within the LTRXLIT command you will be shown a list of DDS source members that use the literal you are currently referencing:

```

LTX510D1                Display Literal Usage
Select  Language: 2929   German Translations System: INVT

Literal ID: INV0964  Order Number:

Type options, press Enter.
  1=Display source member  2=Display screens/Print report

   Source      Source      Member
Opt Library    File         Name         Text Description
-----
LTF           QDDSINVT   SAB010D     Create Supplier Orders
LTF           QDDSINVT   SAOE55D     Credit Note Entry
LTF           QDDSINVT   SAOE63P     Repricing Audit Report
LTF           QDDSINVT   SAOE68D     Credit Sundry Charges Maintenan
LTF           QDDSINVTL  SAOE70D     Order Entry Function
LTF           QDDSINVT   SAOE70D     Order Entry Function
LTF           QDDSINVT   SAOE72D     Sundry Charges Maintenance
LTF           QDDSINVT   SAPO58D     Purchase Order Maintenance
LTF           QDDSINVT   SAPO81D     Purchase Order Inquiry by Order

                                                    Bottom

F3=Exit  F12=Previous  F18=Work with spool files

```

If you wish to see the actual DDS source code for any of the members listed, select the member by typing a '1' beside it and press ENTER. This will then cause the actual (converted) DDS source for the member to be displayed (for control functions of this display screen, refer to the iSeries manual relating to using the STRSEU command²).

If you wish to show the layout of the screen, or print the layout of the report, select the member by typing a '2' beside and press ENTER. This will then use IBM facilities to either display the screens (via SDA³), or print a 'prototype' report (via RLU⁴). Note that, for this feature to work, you must have the IBM Application Development Tools program product (product code 5769-PW1) installed on your iSeries.

The F18 function key option will allow you to view any prototype reports that you have generated from this feature.

² IBM Source Entry Utility Manual. Refer chapter 3, 'Browsing Members'

³ For a discussion of the SDA 'test' function, refer to the IBM Screen Design Aid Manual, chapter 6, 'Testing Display Files'.

⁴ For a discussion of the RLU 'prototype' report feature, refer to the IBM Report Layout Utility Manual, chapter 3, 'Printing a report prototype'.

Searching for the Usage of a word in a literal

If you use the *F16=Search Literals* function key within the LTRXLIT command you will be shown an enquiry screen allowing you to perform a keyword search on all literals defined in the system you are currently accessing:

```

LTX501D1                               Word Search on Literal
Select  Language: 2929   German Translations System: INVT

Type options, press Enter.
5=Display

Keyword          Literal  Literal Text
NUMBER           INVT0964  Order Number:
ORDER            INVT0964  Order Number:
SEL              INVT5674  Sel Sundry Charge to apply      Char
SELECT           INVT4667  ?=Select
STAND            INVT2722  STAND
SUNDRY           INVT0958  Sundry Charges
SUNDRY           INVT5674  Sel Sundry Charge to apply      Char
TAX              INVT5674  Sel Sundry Charge to apply      Char
TESTED           INVT7529  Time:
TIME             INVT0021  Time:
TO               INVT5674  Sel Sundry Charge to apply      Char
VALUE           INVT2724  Charge Value

F3=Exit  F12=Previous  F13=Switch to Translation Search

```

By entering a word on the 'Select key' prompt you will position the list to the usage of that word within all literals. To display further detail for one of the literals shown you can select it by typing a '5' beside it and pressing ENTER.

You will then be shown this detailed information for each literal selected on the previous screen:

```

LTX501D                               Word Search on Literal
Display Language: 2929   German Translations System: INVT

Literal ID.....: INVT0021

Literal Length.....: 5
Number of uses.....: 52

Original Literal Text:
Time:

Translation:
Zeit:

F3=Exit  F12=Previous  F13=Switch to Translation Search

```

Searching for the usage of a word in a Translation

If you use the *F17=Search Translations* function key within the LTRXLIT command you will be shown an enquiry screen allowing you to perform a keyword search on all translated literals defined in the system you are currently accessing:

```

LTX502D1                Word Search on Translation
Select  Language: 2929   German Translations System: INVT

Type options, press Enter.
5=Display

Opt Translated keyword  Literal  Translation Text
AB
AB                    INV5857  <Ab          Teilenummer          Preis
AB                    INV5880  G~ltig ab:
AB                    INV6139  Folgende R~ckstandsorder wurden reserviert ab
AB                    INV7406  Nummer Reihe ab....:
ABBRECHEN             INV0143  - Anfrage abbrechn
ABBRECHN              INV7033  F2=Order|ffng abbrechn
ABBRUCH               INV0246  EINGABE AKZEPTIERT; Y -> WEITERFAHREN, N -> A
ABER                  INV2342  2- Kredit Result. eingegeb. aber ni
ABER                  INV2344  3- GU-Resultat eingegeben aber GU
ABER                  INV3588  1- Adresse eingegeben aber keine
ABF                   INV0373  ALPHA LIEFERANTENSTAMM ABF.
ABF                   INV3265  Drucker Datei~berschreib. - Abf.

F3=Exit  F12=Previous  F13=Switch to Literal Search

```

By entering a word on the 'Select key' prompt you will position the list to the usage of that word within all translations. To display further detail for one of the translations shown you can select it by typing a '5' beside it and pressing ENTER.

You will then be shown this detailed information for each literal selected on the previous screen:

```

LTX502D                Word Search on Translation
Display Language: 2929   German Translations System: INVT

Literal ID.....: INV7406

Literal Length.....: 20
Number of uses.....: 1

Original Literal Text:
Number range from..:

Translation:
Nummer Reihe ab....:

F3=Exit  F12=Previous  F13=Switch to Literal Search

```

Correcting Words in the Translations

If you chose the *WRDCOR function of the LTRXLIT command, you will access the Word Correction function. This facility allows you to display the usage of a word and correct any - or all - of the translations using it. The first panel displayed lists all of the words defined within all literals:

```

LTX503D1                Translated Word Correction
Select  Language: 2929   German Translations System: INVT

Type options, press Enter.
1=Select

Opt Keyword              Opt Keyword              Opt Keyword
ACC                      ORDER
ACC                      SEL
APPLY                    SELECT
BRANCH                   STUDY
CHARGE                   SUNDRY
CHARGES                  TAX
CODE                     TESTED
COMMANDS                 TIME
DATE                     TO
DAILY                    VALUE
F11DELETE                1RETURN
F2PREVIOUS               2RESET
NUMBER

F3=Exit

```

The 'list' area of the screen will show 36 keywords at a time. You can scroll through the keywords by pressing the PAGE UP or PAGE DOWN keys. You can reposition the list to a specific section of the file by entering a keyword in the 'Position list to' prompt and pressing the Enter key.

The information displayed in the list section is as follows:

Opt	Select keyword	If you enter a '1' beside any word, you will select the word for processing (see next section 'Correcting uses of the word').
Keyword		This is a list of all words that have been used in the literals known to LTF.

As previously stated, you can enter '1' beside any word in the list to select it for processing. If you enter a '1' beside more than one word you can process multiple words without having to return back to this screen.

The function keys available on this display are as follows:

F3 Exit from the Literal Translation command.

Correcting uses of the word

The next screen will display a list of all literals that contain the word selected. Where a translation has been made, the translation will appear on the screen. Where no translation has been made, the original literal will be displayed:

```

LTX504D1          Translated Word Correction
Select Language: 2929 German Translations System: INVT
Type options, press Enter.
  1=Select  2=Change  5=Reset  Current Translation...: CODE
                                     Enter Correction.....: CODE
Opt Literal      Translation
RWH0366
RWH0366         Vorgegeb. Versand Code:
RWH0726         Lager Code
RWH0754         Lager Code :
RWH1778         Lager Code.....:
RWH2439         Versand Code.....:
RWH3382         Lieferanten Code...:
RWH3431         <Teile Nummer          Datum          Preis          Code
RWH3459         <Teile Nummer          Methode          Lagerungs Bemerk.
RWH3473         Rabatt Code.....:
RWH3496         Versand Code.:
RWH3550         - Eingeben g~ltig. Schl~sselkodes. M|gliche Varianten
RWH3563         -> g~ltige Lager Code.
RWH3667         Versand Code / Beschreibung          Zeit pro Zeile          pro

F3=Exit  F12=Previous

```

The 'list' area of the screen will show up to 13 literals at a time, each literal containing the requested keyword. You can scroll through the literals by pressing the PAGE UP or PAGE DOWN keys. You can reposition the list to a specific section of the file by entering a literal ID in the 'Position list to' prompt and pressing the Enter key.

The information displayed in the list section is as follows:

Opt	Select literal	<p>If you enter a '1' beside any word, you will select the literal to have word replacement performed. Word replacement will take the entry you made at the top of the screen (the 'Enter Correction' prompt) and replace it in the selected literals. The replacement will be made for each literal selected. See <i>Notes</i> below.</p> <p>If you enter a '2' beside any literal you will be given a window in which to alter the complete text of the translation for that literal. See the next section of this manual 'Translation Correction Window' for further details.</p> <p>If you enter a '5' beside any literal, the literal status will be reset to 'Not Yet Translated' and the text will return to the original text (the translation will be removed).</p>
Literal ID		This is the unique identifier for a literal.
Sts	Status	<p>This is the status of the literal in regards to its translation. The entry in this column may be one of the following:</p> <ul style="list-style-type: none"> '****' Not yet translated 'IGN' Translation ignored 'RPL' Literal is being replaced ' ' Translation entered

Translated text

The first 60 characters of the literal text. An arrow delimiter '<' at the beginning of the literal signifies that the literal has leading blanks and the literal has been 'shuffled up' so that the first non-blank character in the literal is displayed. This function is purely for display purposes. The actual literal remains unchanged.

The text shown will be the current translation. If no translation has yet been entered, the original text will be displayed.

You can enter a selection option beside any literal in the list to select it for processing. If you enter a selection option beside more than one literal you can process multiple literals without having to return back to this screen.

Notes:

Current translation: Enter in this field the character string that you want corrected. This is case-sensitive information and only an exact match will be considered.

Enter translation: Enter in this field the replacement character string. Where an exact match is found in a selected literal for the 'current translation' string, it will be replaced with the character string entered in this field.

The function keys available on this display are as follows:

F3	Exit from the Literal Translation command.
F12	Ignore processing of this keyword.

Translation Correction Window

If you used literal selection option '2' on the previous screen, a 'window' will be displayed on the screen allowing you to change the translation text for that specific literal:

```

LTX504D1          Translated Word Correction
Select Language: 2929   German Translations System: INVT
Type options, press Enter.
  1=Select  2=Change  5=Reset  Current Translation...: CODE
                                           Enter Correction.....: CODE

Opt Literal      Translation
RWH0366
RWH0366   Vorgegeb. Versand Code:
RWH0726   Lager Code
RWH0754   Lager Code :
RWH1778   Lager Code.....:
RWH2439   Versand Code.....:
 2 RWH3382   Lieferanten Code...:
RWH3431   <Teile Nummer          Datum          Preis          Code
RWH3459   .....:
RWH3473 :           Change Translation Text           :
RWH3496 : Lieferanten Code...:           :
RWH3550 :           :           :
RWH3563 :           F12=Cancel :
RWH3667 : .....:

F3=Exit  F12=Previous

```

This window will display the current translation for the selected literal. If no translation has yet been made, the original text will be displayed.

If selection option was entered beside more than one literal, this window will be displayed for each literal that was selected.

The function keys available on this display are as follows:

F12 Ignore processing of this literal

Building/Rebuilding the Translation Environment

It is extremely unlikely that system failures will cause the language library to be damaged. However, if it ever happens you need not worry - the library can be completely rebuilt using the Rebuild Translation Environment (LRBLTRXENV) command.

This command is also useful for generating an initial (untranslated) language environment.

```

                                Rebuild Trx Environment (LRBLTRXENV)

Type choices, press Enter.

Language to process . . . . . LANGUAGE      > 2929
System to process . . . . . SYSTEM         > INVT
Rebuild literals message file? RBLITMSGF   *YES
Recreate translated objects? . . RBLTRXOBJ  *YES
Job description . . . . . JOB              > LTFJOB
  Library name . . . . .                  > LTF
Output queue . . . . . OUTQ               LTFOUTQ
  Library name . . . . .                  LTF

```

This command will rebuild the Literals Message file, all word cross reference indexes, all source member reference indexes and then recreate all printer and display file objects - for a specified system and a specified language.

Rebuild literals message file?: This specifies whether the job will attempt to rebuild (delete, and then re-create) the translated literals message file based on the information currently held in the literals translation database.

Recreate translation objects?: This specifies whether the job will attempt to rebuild (delete, and then re-create) all translated objects (display and printer files) for the specified system in the specified language library.

Each object will be submitted for individual creation. The processing may run for a long time, dependent upon the number of objects to be re-created.

Notes on message file rebuild

When you use the LRBLTRXENV command, the literals message file in the translation library will be completely rebuilt based on the current translation status of the system.

Each literal defined to this system will be added to the literals message file in the LTFaaaaaaa translation library, using the following rules:

- If a translation exists, the translation will be loaded
- If no translation has yet been entered, the original (untranslated) literal will be loaded.

As such, the result of this rebuild may be that the display and printer file objects contain a mixture of translated and untranslated literals.

Creating individual Display and Printer file objects

A generic (or bulk) object creation utility, Create Object (NCRTOBJ⁵) has been included with LTF to allow you to create all converted DDS members with one 'create' command, rather than having to submit each one individually:

```

                                Object Creation Facility (NCRTOBJ)

Type choices, press Enter.

Source file name . . . . . SRCFILE      > QDDSINVT
  Library name . . . . .                >   LTF
Member name . . . . . MBR                > *ALL
Object library name . . . . . OBJLIB     > LTF2929
Source member attribute . . . . . ATTR   > *ALL
Job description name . . . . . JOBDB     > LTFJOBDB
  Library name . . . . .                >   LTF
Output queue name . . . . . OUTQ        > *CURRENT
  Library name . . . . .                >   *LIBL
Authority . . . . . AUT                  > *LIBCRTAUT

```

The prompts have automatically been filled in for you, based on the system and language you are currently accessing.

The command, as supplied, will submit the object creations using the job description defined. The library list for the submitted jobs is *CURRENT, so please ensure that your current library list is correct before requesting this option.

Your current library list must contain the names of all libraries referenced within the source members being processed, as well as the LTF libraries and the QTEMP library.

⁵ The NCRTOBJ command is normally a function of the NUTIL (*Navan Utilities*) program product library.

Rebuilding the Word Indexes

Option 6 on the LTF menu allows you to completely rebuild the keyword cross reference files for the current system/language in use:

```

                                Submit Job (SBMJOB)

Type choices, press Enter.

Command to run . . . . . CMD          > CALL PGM(LRB010)
Job name . . . . . JOB                > aaa_RBLDIN
Job description . . . . . JOBBD       > LTFJOBBD
  Library . . . . .                   > LTF
Job queue . . . . . JOBQ              *JOBQ
  Library . . . . .                   *JOBQ
Output queue . . . . . OUTQ           *CURRENT
  Library . . . . .                   *CURRENT

```

The job name for the submitted job will be `_RBLDIN`, prefixed by the identifier of the current system in use.

Printing a list of Literals/Translations

The `LPRTLIT` command (option 7 on the menu) will print a list of Literals/Translations within the criteria you enter on the command prompt:

```

                                Print Literals/Translations (LPRTLIT)

Type choices, press Enter.

Report type . . . . . RPTTYP          *REPORT
Listing type . . . . . LSTTYP        *ALL
From literal . . . . . FROMLIT       *FIRST
To literal . . . . . TOLIT           *LAST

```

The command parameters are as follows:

- | | |
|--------------------|---|
| Report type | The type of report you require:
*REPORT gives a basic list
*WORKSHEET provides space below each literal for you to be able to write in your translations/notes. |
| List type | The type of list to be generated:
*ALL prints all literals/translations
*TRX prints all literals that have been translated
*NOTRX prints all literals that have not yet been translated
*IGN prints all literals that have been ignored for translation |
| From/to literal ID | Allows you to restrict the report to a specific group of literals |

Moving other 'text' objects into the Translation library

There may be other application 'text' objects that will require translation. Examples of these are Error Messages, Parameter Descriptions, Menu Options and so on.

These require a separate means of translation. In most cases the best way to translate these is to create a copy of the object in the Translation Library and then modify any text manually.

The LTF menu assists you in this phase by providing three menu options:

- Option 10 - Create a copy of an object into the Translation Library
- Option 15 - Modify a Message File
- Option 16 - Modify a Data File

The functions behind these menu options are standard OS/400 commands CRTDUPOBJ (create Duplicate Object), WRKMSGD (Work with Message File Descriptions) and UPDDTA (Update a Data File) respectively.

Create a copy of an object into the Translation Library

Option 10 of the LTF menu performs the 'Create Duplicate Object' command:

```
                Create Duplicate Object (CRTDUPOBJ)

Type choices, press Enter.

From object . . . . . OBJ
From library . . . . . FROMLIB
Object type . . . . . OBJTYPE
                + for more values
To library . . . . . TOLIB      > LTF2929
New object . . . . . NEWOBJ     > *SAME
```

The Create Duplicate Object command will duplicate a specified object into the Translation Library. This command can be used for any object and object type on the system, but you must have authority to perform the function.

If you are taking copies of database files that have associated logical views, you must copy the Physical Database file first and then duplicate any required logical views over it. If you do not do this it may cause unpredictable results when running the application.

Some of the Command prompts have automatically been filled in for you, based on the system and language you are currently accessing.

Should you require information on using the CRTDUPOBJ command, refer to the IBM OS/400 CL Reference Manual.

Entering translations into a Message file

Option 15 of the LTF menu performs the 'Work with Message Descriptions' command:

```
Work with Message Descriptions (WRKMSGD)

Type choices, press Enter.

Message file . . . . . MSGF          > MSGF
Library . . . . .                > LTF2929
```

The Work with Message Descriptions command allows you to change the text associated with one or more messages in a Message File object. It is advisable to take a copy of the original Message File object (using LTF menu option 10) and work on the copy, rather than modifying the original object.

This command **must not** be used to alter the LTF Literals Message File directly, otherwise unpredictable results could occur when creating Display or Printer files that access the translations.

Some of the Command prompts have automatically been filled in for you, based on the system and language you are currently accessing.

Should you require information on using the WRKMSGD command, refer to the IBM CL Reference Manual.

Entering translations into a Database file

Option 16 of the LTF menu performs the 'Update Data' command:

```
Update Data with Temp Program (UPDDTA)

Type choices, press Enter.

Data base file . . . . . FILE      > FILE
Library . . . . .                > LTF2929
Member . . . . . MBR              *FIRST
```

The Update Data command (which is part of the OS/400 Data File Utilities program product) allows you to change information in any externally defined Database file.

It is advisable to take a copy of the original Database File object (using LTF menu option 10) and work on the copy, rather than modifying the original object.

Some of the Command prompts have automatically been filled in for you, based on the system and language you are currently accessing.

Should you require information on using the UPDDTA command, refer to the appropriate IBM ADTS Data File Utility User Manual.

Working with converted source

Option 17 of the LTF menu performs the 'Work with Source Members' (WRKMBRPDM) command, if it is installed on your machine.

This allows access to the source DDS that has been converted for the system you are currently working in.

It is recommended that you do not work on converted source. It is preferable to alter the original DDS source member and then use the LCVTDDS command to convert it back into LTF.

Accessing the IBM Programmers Menu

Option 20 of the LTF menu performs the 'Start Programmers Menu' (STRPGMMNU) command:

```

                                Programmer Menu
                                System:  NAVAN

Select one of the following:
  1. Start iSeries Data File Utility
  2. Work with iSeries Query
  3. CALL LTF/NCRTOBJECT          object name, type, pgm for CMD
  4. Call a program              program name
  5. Run a command               command
  6. Submit a job                (job name), , ,(command)
  7. Go to a menu                menu name
  8. Edit a source file member   (srcmbr), (type)
  9. Design display format using SDA (srcmbr), ,(mode)
 90. Sign off                    (*nolist, *list)

Selection . . . . . Parm . . . . .
Type . . . . . Parm 2 . . . . .
Command . . . . .

Source file . . . . . QDDSINVT      Source library . . . . . LTF
Object library . . . . . LTF2929    Job description . . . . . LTFJOB

F3=Exit      F4=Prompt      F6=Display messages  F10=Command entry
F12=Cancel   F14=Work with submitted jobs  F18=Work with output

```

Some of the menu prompts have automatically been filled in for you, based on the system and language you are currently accessing.

You will notice that option 3 of the menu uses the Navan exit program, NCRTOBJECT. This functions in exactly the same way as the normal option 3 but performs additional processing necessary for some LTF functions. NCRTOBJECT is included in the LTF program library.

Should you require information on using the IBM Programmers Menu, refer to the IBM CL Programming Manual.

Appendix

Accessing LTF from within the PDM utility

If you are intending to use the IBM Programming Development Manager product (PDM) from an LTF user profile, you should ensure that you correctly initialize it before attempting any LTF functions:

1. From a command line, type STRPDM and press Enter. This will start the PDM utility. When the main selection panel appears, press F18 to change your defaults
2. On the 'Change defaults' panel you should type in the name of the LTF *nnnn* target language library
3. If you are running the NUTIL (Navan Utilities) program product you should specify the Option file as QAUOOPT in library NUTIL (member QAUOOPT). Otherwise leave the value shown.
4. Press Enter. You have now initialized PDM for this LTF user.

If you are not running NUTIL, you should add the following 'create object' PDM option to your current PDM options file:

Option: CO
Command: ?LTF/NPDMCRTOBJ ?*MBRNAM(&N) ?-MBRTYP(&S) ?*SRCFILE(&L/&F)
??OBJLIB(&O) ??JOBID(&J) ?-LSTTYP(&B)

To add this to your options file, take LTF menu option 17. When the PDM member list panel is displayed, press F16. You will then be shown the 'Work with User Defined Options' panel and your current options will be listed. Press F6 to add a new option and add the above information. Please make sure that you type the command *exactly* as shown above.

General Notes on Translating Applications

Use of special characters

It is advisable to avoid using any 'special' characters in your user displays. This is because the different language keyboards and displays support special characters differently. When you type an ampersand character on your English display it is not necessarily displayed as an ampersand when displayed in another country. To be absolutely safe, it is better if you limit the characters in your display and printer files to the following:

A to Z (and a to z)
0 to 9
+ - . , : ; _ () ? = * ' " > <

All characters other than these are not necessarily supported by all national language character sets.

Abbreviations

Abbreviations should be avoided wherever possible. Whilst you can abbreviate 'number' in English to 'no', it is very difficult in some other languages to abbreviate the same word into a 2 character abbreviation. You should leave as much space as possible when defining a literal string to enable a more accurate translation.

It will be much easier for a translator to convert this screen:

```
Part Number . . . . . :
Description . . . . . :
Supplier . . . . . :
Bin Location . . . . . :
```

than to convert this screen:

```
Part No:           Description:
Supplier:          Bin Locn:
```

due to the amount of space available to enter the translation.

Similarly, when defining subfile and report column headings allow as much space as possible for each heading:

```
'Part Number'  'Description'      'Supplier'  '
```

or even

```
'Part Number'      Description          Supplier      '
```

rather than

```
'Part Number'      'Description'      'Supplier'
```

Literals in arrays and program variables

LTF cannot process literals that have been coded into programs - it only works on DDS. If you have displayable text that is defined within a program you must either provide a translated version of the program or alter the program logic to retrieve the text from a message file. While this sounds complex, it isn't...

Take the following RPG code:

```

*
* Get job status
C          SHSBM      IFEQ 'Y'
C          MOVEL 'Submit' SHSTS  P
C          ELSE
C          MOVEL 'Ready'  SHSTS  P
C          ENDIF

```

You can perform the same function by retrieving the text from a message file as follows:

```

*
* Get job status
C          SHSBM      IFEQ 'Y'
C          CALL '@RTVMSG'
C          PARM 'LIT0050' NMSGID 7          'Submit'
C          PARM 'RPGLIT'  NMSGF 10
C          PARM          NMDTA 80
C          SHSTS      PARM          NMSG 80
C          ELSE
C          CALL '@RTVMSG'
C          PARM 'LIT0051' NMSGID 7          'Ready'
C          PARM 'RPGLIT'  NMSGF 10
C          PARM          NMDTA 80
C          SHSTS      PARM          NMSG 80
C          ENDIF

```

The program used, @RTVMSG, is included for your use in the LTF library (you can copy it to user libraries if required). The program retrieves the message ID requested (NMSGID) from the specified message file (NMSGF), inserts the message data (NMDTA if any) and returns the formatted message (NMSG) back to your program.

In the above example, a message file RPGLIT was previously created and has two message identifiers in it:

```

ADDMSGD MSGID(LIT0050) MSG('Submit')
ADDMSGD MSGID(LIT0051) MSG('Ready')

```

Characters for options and parameters

Where possible, use numbers rather than characters for selection options. While option 'C' to copy is meaningful in English, it does not make much sense to a Norwegian, who has to do a Kopi. If you use a number as the option character

'3=Copy '

the resulting translations can become equally understandable in all languages.

Similarly, be careful with optional parameter values that the user can enter. For example, if you require the user to respond to a prompt by entering *YES or *NO you are complicating things for a French user, who would naturally expect the valid values to be *OUI or *NON.

Date Formats

Different countries use different date formats and it is unreasonable to expect an application user in France to enter a date in Month/Day/Year format when they normally work in Day/Month/Year.

A standard approach (used by Navan and many other companies) is that all dates are stored in the database in Year/Month/Day format. These are then displayed/entered by the user in Job format (which can be determined from the user's job attributes via the RTVJOBA command).

Another date-related problem to consider is that the traditional date editing code EDTCDE(Y) does not work on 8 digit dates. So when you are deciding how to display a full 8 digit date on the display or report you need to bear in mind that the editing format will change depending on where your application is being used:

Sweden	2002/06/15
USA	06/15/2002
England	15/06/2002

As you can see, the date separator character 'moves' within the edit mask. As such one edit code, or edit word, is insufficient for displaying an 8 digit date.

Date handling routines

In order to help you with developing applications that support international date formatting, there are three callable utility programs supplied with LTF. These are explained in detail on the following pages.

@RTV DAT8 - Retrieve the current system date/time

This program will retrieve the current system date and time and return the date in *ISO format.

Parameter List

<u>Field</u>	<u>Defn</u>	<u>Usage</u>	<u>Description</u>
@Date8	8P0	O	Field containing the date in YYYYMMDD format
@Time	6P0	I/O	Field containing the time

EXAMPLE OF USAGE

```
* ...Retrieve system date in *ISO format....  
C          Call      '@RTV DAT8'  
C          Parm          @Date8  80  
C          Parm          @Time   60
```

@CHKDAT8 - Date validation routine

This program will test a given date for validity. If the date passed to the routine is invalid, a value of 99999999 will be returned as the date. Note that the program remains open in the default activation group after the first call, to improve response.

Parameter List

<u>Field</u>	<u>Defn</u>	<u>Usage</u>	<u>Description</u>
@Chkd8	8P0	I/O	Field containing the date
@Dfi8	4A	I	Date format in, either *DMY Day/Mth/Year *MDY Mth/Day/Year *YMD Year/Mth/Day *ISO System format *EUR EUR format *JIS JIS format *JUL JUL format *USA USA format *JOB Job format

Note that the date passed must be valid date data within standard rules. A zero date is considered invalid.

EXAMPLE OF USAGE

This code will test field ScreenDate to see whether it contains a valid date in the current *JOB format. If it fails the check then the program will send an 'Invalid date' message to the user.

```

* ...Check Date is valid (only if entered)...
C   ScreenDate   Ifne   0
C   ScreenDate   Andne   999999
C                   Call   '@CHKDAT8'
C                   Parm   ScreenDate   @Chkd8
C                   Parm   '*JOB'       @Dfi8

* ...Date does not conform to Job Format...
C   @Chkd8       Ifeq   99999999
C                   Move   *ON           *IN98
C                   Move   'GEN0061'     Nmsgid
C                   Call   'QMHSNDPM'     Smlst
C                   Endif
C                   Endif

```

@RVSDAT8 - Convert date from one format to another

This program will reformat a given date, from one date format to another. The input date will then be returned to the requesting program in the requested format.

If the date passed to the routine is invalid for the format specified, a value of 99999999 will be returned as the date.

A date of 0 is an invalid date.

A date of 99999999 will be ignored by the program.

Parameter List

<u>Field</u>	<u>Defn</u>	<u>Usage</u>	<u>Description</u>
@Date8	8P0	I/O	Field containing the date
@Dfi8	4A	I	Date format in, either *DMY Day/Mth/Year *MDY Mth/Day/Year *YMD Year/Mth/Day *ISO System format *EUR EUR format *JIS JIS format *JUL JUL format *USA USA format *JOB Job format (6 digits) *JB8 Job format (8 digits)
@Dfo8	4A	I	Date format out, either *DMY Day/Mth/Year *MDY Mth/Day/Year *YMD Year/Mth/Day *ISO System format *EUR EUR format *JIS JIS format *JUL JUL format *USA USA format *JOB Job format (6 digits) *JB8 Job format (8 digits)

NOTES ON 8 DIGIT DATES

The century portion of the date is implied for date formats *DMY, *MDY and *YMD, as follows:

- Years in the range 40-99 will be treated as 1940-1999
- Years in the range 00-39 will be treated as 2000-2039
- 6 digit dates outside of the range 1940-2039 are invalid dates

The century portion of the date is a required entry for date formats *ISO, *USA, *EUR, *JIS. If not entered, the century portion will be treated as 00.

EXAMPLE OF USAGE

This code will convert the contents of the FileDate field from *ISO to *JOB format and move the result into the field ScreenDate.

```
* ...Reverse date from *ISO format to *JOB format...
C   FileDate      Ifne      0
C   FileDate      Andne     99999999
C           Call      '@RVSDAT8'
C   ScreenDate    Parm      FileDate      @Date8
C           Parm      '*ISO'          @Dfi8
C           Parm      '*JOB'          @Dfo8
C           Else
C           Z-add     0              ScreenDate
C           Endif
```

The date separator character

The character used as the date separator character is not considered in the above utility programs and will be displayed as whatever the system value QDATSEP was set to when the Display file or Printer file was created.

You should therefore ensure that this System Value is defined correctly before you create your Display and Printer File objects. Use the DSPSYSVAL QDATSEP command to see what the value is for your system.

If you are intending to load your application on to another machine where the date separator character is different, you must re-create all of your display and printer file objects to pick up the new separator character.

If it is necessary to have a variable date separator character then you must code for this manually.

Currency formats

The format of a monetary amount varies widely and the length of the field should be a consideration when developing new applications. If you are developing software to be used in the USA you might design a monetary value field to be 11 positions long with 2 decimal positions. In Mexico you may use 13 positions long, no decimal positions. If your application has a monetary value field defined as 5 positions with 2 decimals it is almost unusable in a country that uses Pesetas.

Whatever the requirement, you should take into consideration that some currencies do not have decimal positions (Yen, Lire etc) and some cannot do without them (Dollars, Pounds, Euro).

The currency symbol, if it is to be displayed anywhere, should be retrieved from the system value QCURSYM.

The character used as the decimal marker is also a consideration. In Dollars the decimal marker is a '.', but in Kroner the decimal marker is a ','. If this is a consideration you must process the system value QDECFMT.

If you have a routine to convert an amount into an alpha equivalent, it will not work across languages. A good example of this type of routine is a program to convert a cheque amount into an alpha string for printing on a cheque:

125.25 converts to

'ONE HUNDRED AND TWENTY FIVE DOLLARS AND TWENTY FIVE CENTS'

which, of course, would be a meaningless term on a financial document in France.

Printer files

Do not assume that all countries use the same size page lengths; in the USA a page size of 66 lines is in general use, but in Norway the standard page size is 48 lines. Your software should be able to cater for this so that the only requirement in moving your application from one country to another is to perform a Change Printer file command:

```
CHGPRTF FILE(LTF/*ALL) PAGESIZE(48) OVRFLOW(44)
```

This means that no program logic should be controlling either the page length or the overflow line number.

Error Messages

All error messages should be external from the program, contained in message files. If you have error messages contained in arrays or variables in your programs then your programs must be changed to process the messages from message files via one of the following methods:

- On a display format you can use the ERRMSGID keyword, conditioned by an indicator that is conditioned by the program when the relevant error occurs.
- On a subfile you can use the SFLMSGID keyword, conditioned by an indicator that is conditioned by the program when the relevant error occurs.
- You can define a message subfile in your Display file that your program will send error messages to.

Help Text, User Guides and Documentation

The conversion of documentation is a manual process that cannot be easily automated - they must be translated word by word. However, PC products are available on the market that can assist with text translation.

The complexity of converting help text depends on how your application processes the 'help' key function. If it is program dependent (it is contained in an array in the program) you should consider changing your program to 'externalize' the text:

- If it is in a source file member, conversion can be achieved using the PDM, SEU or EDTTXX functions of OS/400.
- If it is an Office Help Document (HLPDOC) it can be translated using the WRKDOC function of OS/400.

Utility Commands

Using the LCRTDDS command to convert an Internally Described Printer File

As previously stated, the MSGCON/MSGID keywords are available in DDS only and therefore all Printer Files and Display files must be externally defined from the programs that use them.

If you have legacy source code originally developed on the IBM System 34, System 36 or System 38 systems, it is possible that you may still have Internally Defined printer files in your application. This command allows the conversion of these internal descriptions into corresponding external descriptions.

The LCRTDDS command is *not shown* on the LTF menu; you must request it from a command entry line.

The command will not be processed interactively; if requested from a command entry line it will automatically submit itself for processing after the user has pressed enter.

```
                Create PRTF source (LCRTDDS)

Type choices, press Enter.

From RPG source file . . . . . FROMRPG          QRPGSRC
  Library name . . . . .                   *LIBL
From source member . . . . . FROMMBR
To RPG source file . . . . . TORPG             QRPGSRC
  Library name . . . . .                   *LIBL
To DDS source file . . . . . TODDS            QDDSSRC
  Library name . . . . .                   *TORPGLIB
```

General Overview

The LCRTDDS command allows the modification of an RPG program that contains program described printer files, to use externally described printer files.

- The DDS source for the printer file(s) required is automatically generated in the source file specified.
- An RPG source member will also be generated (in the source file specified).

This new RPG source member will be based on the old member, but will have additional RPG program code included to process the new externally described Printer File.

In most cases you will need to access the new source to perform some modifications before attempting to compile it. The extent of these manual changes will vary, depending on how complex your Printer output is (refer to *Command Limitations*, later in this section).

Command Parameters

From RPG source file:	The name of the source file containing the original RPG source member
From source member:	The RPG Source Member Name
To RPG source file:	The name of the source file that will contain the new RPG member (this must be different to the <i>from file</i>)
To DDS source file:	The name of the source file that will contain the generated Printer File source member. The default value of *TORPGLIB will place the newly created source member into the specified source file in the same library as the RPG source file.

Processing Requirements

The LCRTDDS command will create Externally Defined DDS source, and any fields within the DDS will be referenced back to the data files upon which they are based.

The way this works is that LCRTDDS will first access the File Specifications of the program and for each Externally Defined file it will retrieve that files Database Attributes. If external descriptions for an output field cannot be found then the program will be scanned for an definition of the field. The length and data type of the field will then be set accordingly. If no reference to the field can be found then the DDS for the field will show a field length of '000', data type 'A'.

When you run the LCRTDDS command you must ensure that your library list contains references to *all libraries* necessary to access *all files* referred to in the RPG program's File Specification section.

Command Features

This command will cater for the following RPG output specification functions:

- Edit codes
- Edit words
- Space Before & Space After
- Skip Before & Skip After
- UDATE
- PAGE*n*
- Output Conditioning Indicators

Equivalent DDS specifications will be generated to perform these functions.

Command Limitations

EDIT WORDS

LCRTDDS will generate an edit word equivalent to one defined on the output specification line. Be aware that the RPG Compiler allows edit words to be longer than the field they are editing, but DDS does not.

When you attempt to create a Printer File from the generated DDS an error will occur wherever the edit word is longer than the field it is editing. To correct this problem you must manually change the edit word in DDS and adjust the starting position of the field (since the overall length of the field will be smaller).

ARRAYS

i). When a field on the output specification is an array with an index (for example, ARY,X) then the LCRTDDS command generates a MOVE calculation statement in the RPG program to move the array element to a newly created 'output' field. This MOVE statement must be repositioned manually, to the appropriate location in the RPG program (that is; before the EXCPT statement for E type output, at the end of the detail calculations for D type output and at the end of the Total calculations for T type output).

ii) When an array is specified with no index then all the elements for the array are printed across the page. The LCRTDDS command cannot generate an equivalent operation in DDS so each element of the array must be moved to separate work output fields in the RPG program (this may be done by moves or a data structure) which must then be included on the DDS for the printer file.

INDICATORS

DDS will only support indicators 01-99. The LCRTDDS command will place any output conditioning indicators in the DDS, however when you attempt to create the file the creation will fail if they are not in the range 01-99 (for example the indicator L1 will not work).

If this condition exists you must manually insert code to process an 'alias' indicator that the DDS can use. A move of the original indicator to the new one must be then located at the correct line in the program (that is; prior to the EXCPT statement for E type lines, at the end of the detail calculation specifications for D type specifications, and after the total calculations for T type calculations).

BLANK AFTER

DDS has no equivalent function to the Blank After feature, so the LCRTDDS command generates a MOVE *BLANKS operation code in the RPG program after the relevant 'O' specification line. These generated lines must be manually relocated to their correct position in the program (that is; after the EXCPT statement for E type output, at the beginning of the detail calculations for D type output, and at the beginning of the total calculations for T type output).

OVERFLOW

The RPG compiler will only allow indicators 01-99 to be specified as overflow indicators on the File Specification for an externally described printer file. If this condition exists, the LCRTDDS command will write a comment after the 'F' specification informing the user that an indicator has been removed from the 'F' specification for the Printer file. An appropriate replacement indicator must be used and all subsequent references to the original indicator in both the RPG and DDS source changed.

FETCH OVERFLOW

Fetch Overflow logic cannot be used in externally described printer files. If Fetch is specified on any printer output it must be manually removed and the appropriate code included in the RPG program to detect for Overflow.

CONDITIONED SKIP/SPACE INSTRUCTIONS

If SPACE B/A (Space Before/After) or SKIPB/A (Skip Before/After) are used on an output specification with an OR line, then the line controls are controlled by the indicators on the respective line. For example:

```
OQSYSPRT  12      21N24
O          OR 101  21 25
```

These two lines of code cause the program to space 1 line before and 1 line after when indicator 21 is on and 24 is off; OR it skips to line 21 before and spaces 1 line after if 21 is on and 25 is on.

When LCRTDDS generates the DDS for this logic as follows:

```
A          R  PRTFMT01
A  21N24      SPACEB(1) SPACEA(2)
AO 21 25      SPACEA(1) SKIPB(01)
```

This causes a DDS compilation error since SPACEA cannot be defined twice for the same record format. The logic for these lines must be manually changed to have the keyword defined only once.

DUPLICATE FIELD NAMES IN THE SAME RECORD FORMAT

DDS will only allow a field to be specified once within any given printer file record format. Therefore any field names that are 'duplicated' will have to be manually renamed and a Data Structure defined (or MOVE operation used) to force an automatic move to the new field.

Translating Literals for a specific source member

During the 'cleanup' phase of a translation, it is usually necessary to change the literals relating to a specific screen or report.

If you know the DDS source member name, you can use the LTRXLIT command to translate specifically for one source member by entering the source member name in the relevant command parameter, as follows:

```
Translate Literals (LTRXLIT)

Type choices, press Enter.

Display sequence . . . . . SEQUENCE      > *LITID
System to translate . . . . . SYSTEM      *CURRENT
Start from literal . . . . . FROMLIT      *START
Source member name . . . . . MBRNAM       > SRA512P
Source file . . . . . SRCFILE            *SYSDFT
Library name . . . . .
```

In the above example, the translation program will display only those literals used in source member SRA512P. The Display Sequence must be *LITID, which is the only valid sequence when translating literals for a specific source member.

Business Contingency Planning

In the event of a serious failure of the CPU for which you have licensed this copy of LTF, it may be necessary to temporarily transfer your business processing to another CPU.

While it is not possible to temporarily transfer your LTF product license to another CPU, it is possible to temporarily run your copy of LTF on another machine. The way to do this is as follows:

Using your most recent backup copy of LTF, restore the library from your offline storage media onto the temporary CPU. Now type the following commands *exactly as shown* on a command line:

```
ADDLIBLE      LIB (LTF)
CHGDTAARA    DTAARA (LTFDTA (17 8))  VALUE (*DCS)
```

Now type CALL LT9000 and press *enter* to display the LTF Installation Attributes panel. Verify that there is no 'copyright violation' error message displayed, and that an expiry date 15 days from now has been established for this temporary copy.

LTF will now function correctly on the temporary CPU. Please note that this copy is installed as a demonstration version and is bound by the terms and conditions of a demonstration version as explained in your license agreement.

Please note that this Disaster Contingency Support option *cannot be extended beyond 15 days*. Any attempt to re-use the *DCS option will result in a copyright violation message and LTF will cease to function.

Once your operations on your main CPU have been re-established you should remove this copy of LTF from the temporary CPU.

Support Contacts

Europe, Middle East and Africa

Navan (UK) Limited
107, The Lakes
Larkfield
Aylesford Kent ME20 6SJ
England

Fax: +44 1622 790520
Email: helpdesk@navan.co.uk

Rest of the World

Navan Australia Pty Ltd
22 Torrington Road
Maroubra
NSW 2035
Australia

Fax: +61 2 9344 9853
Email: helpdesk@navan.com.au

The latest version of this manual, as well as the latest PTF's to this product, can always be downloaded from the Navan Internet Website, which can be found at

<http://www.navan.co.uk>