OCCASION

This publication has been made available to the public on the occasion of the 50th anniversary of the United Nations Industrial Development Organisation.

DISCLAIMER

This document has been produced without formal United Nations editing. The designations employed and the presentation of the material in this document do not imply the expression of any opinion whatsoever on the part of the Secretariat of the United Nations Industrial Development Organization (UNIDO) concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries, or its economic system or degree of development. Designations such as "developed", "industrialized" and "developing" are intended for statistical convenience and do not necessarily express a judgment about the stage reached by a particular country or area in the development process. Mention of firm names or commercial products does not constitute an endorsement by UNIDO.

FAIR USE POLICY

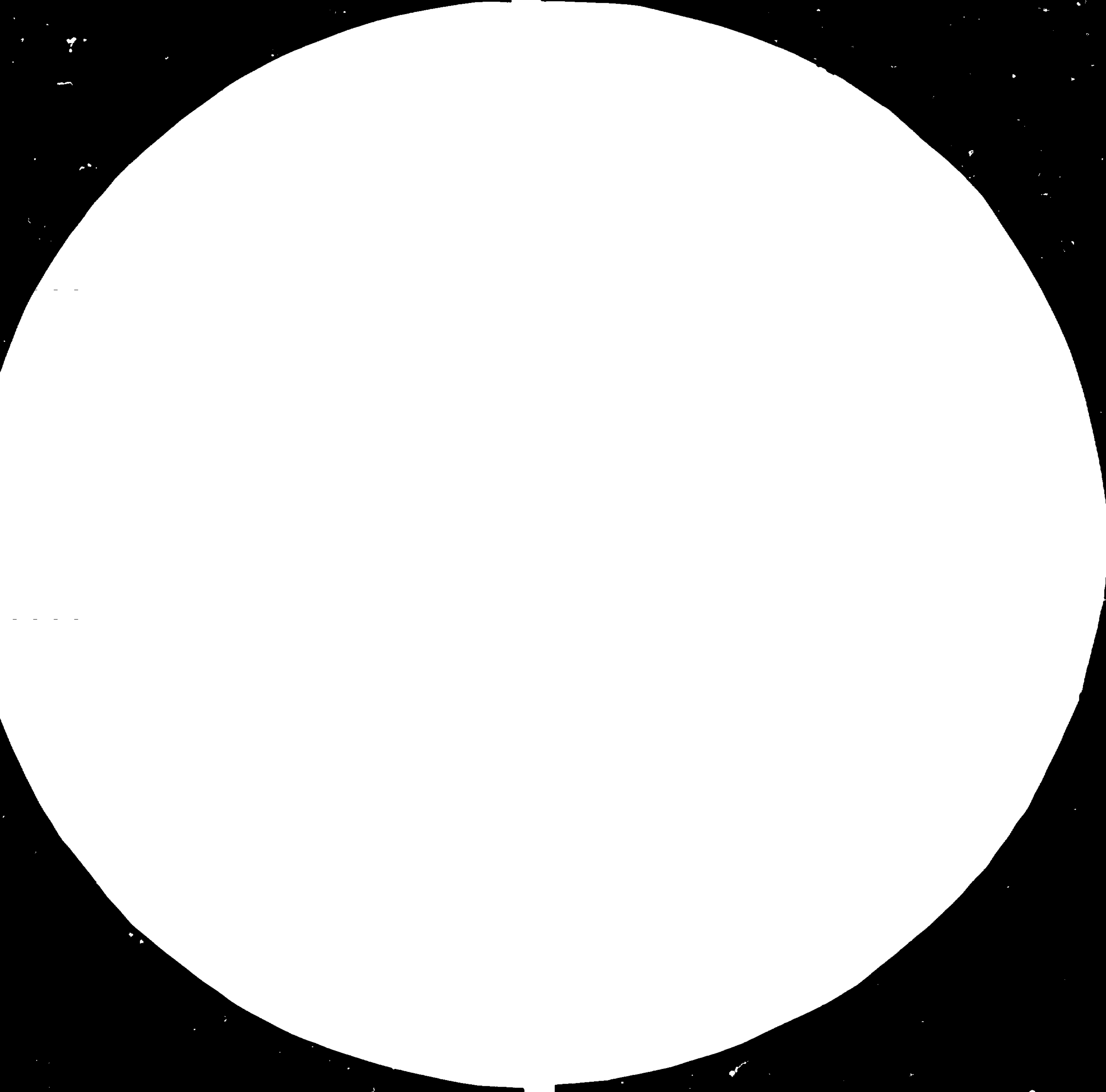
Any part of this publication may be quoted and referenced for educational and research purposes without additional permission from UNIDO. However, those who make use of quoting and referencing this publication are requested to follow the Fair Use Policy of giving due credit to UNIDO.

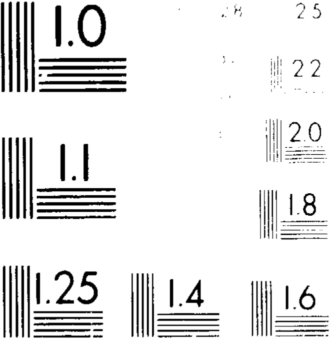
CONTACT

Please contact publications@unido.org for further information concerning UNIDO publications.

For more information about UNIDO, please visit us at www.unido.org

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
Vienna International Centre, P.O. Box 300, 1400 Vienna, Austria
Tel: (+43-1) 26026-0 · www.unido.org · unido@unido.org




1.0
2.8
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

RESTRICTED

DP/ID/SER.B/502
26 April 1985
English

14623

ASSISTANCE TO MACHINE BUILDING INDUSTRY

DP/CPR/79/021

CHINA

Terminal report*

Prepared for the Government of China
by the United Nations Industrial Development Organization,
acting as executing agency for the United Nations Development Programme

China

Based on the work of F.A. Daneliuk and W.S. Page,
experts in MINISIS data base management system design

United Nations Industrial Development Organization
Vienna

* This document has been reproduced without formal editing.

V.85-25496

CONTENTS

INTRODUCTION

This project was carried out in Beijing at the Documentation Centre of the Information Institute of the Ministry of Machine Building. The duties of the consultants were to:

1. train local personnel on data base management system design (especially MINISIS) and on new processor development; and

2. give lectures to local personnel on the design of data base management systems, with emphasis on MINISIS, and on development of applications in MINISIS.

The consultants fulfilled these obligations by:

1. conducting a series of lectures over a twelve day period, to a class of twenty students, on the topics of management information systems, data base management systems (theory and practice), and MINISIS (in-depth exposition);

2. conducting a series of workshops over a period of seventeen days, to a class of ten students, on the topics of theory of structured programming, theory of syntax, development of new processors using MINISIS, development of new applications using MINISIS, and development of special exits for specialized processing in MINISIS;

3. conducting practical hands-on training sessions in the development of new extensions to MINISIS by assisting local staff in the development of a Chinese character processing interface for MINISIS. This interface is now fully operational;

4. providing advice and guidance in the means of using most efficiently the equipment that is available at the Institute. This involved, basically, the development of a technique for transferring Chinese data from the PDP/11 text processing system in use at the Institute, to the HP3000-based MINISIS system;

5. providing assistance in the integration of the new Dragon terminal (which has a limited repetoire of Chinese characters) into the HP3000 system;

6. providing advice on the support of other character sets in MINISIS, character sets which the Institute must support (Cyrillic, Katakana);

7. the development of documentation describing all the work done under this project.

More clarification of these items is contained in the next part of this report. In addition, attachments are provided for further information.

The consultants submit the following recommendations, based on their work with the Institute.

1. Through the successful implementation of the Chinese interface to MINISIS, the Institute is now in a good position to make recommendations to the Government of China on the use of Chinese character processing in data base development.

2. The Institute should undertake to build its data base of Chinese documents/works as quickly as possible so as not to lose the momentum gained through the project.

3. Other MINISIS users in China should be given access to the software developed at the Institute in order to prcliferate the use of Chinese character processing in China.

4. The Government should seriously consider greater funding for the development of Chinese character processing on computer based on software techniques, rather than hardware techniques. The experience of the Institute both in text processing and data base development is positive evidence that software approach is most promising. Software has no artificial restrictions on character generation as do hardware techniques. Furthermore, it is possible with software to minimize the dependence on particular hardware while retaining the flexibility of taking advantage of new hardware as it becomes available.

We understand that a highly respected Chinese scientist, Dr. Qian Wei Chang, at the Shanghai Gongye Daxue University, is working on the software approach to Chinese character processing. The staff of the Institute, in particular Mr. Wu Kang, could provide Dr. Chang with useful input. Dr. Chang, on the other hand, would probably be very interested in the experience of the Institute.

5. The Institute should actively pursue the transfer of data collected through their text processing facility on the PDP/11 system, to a MINISIS data base. (See item 4 in the following pages.)

6. The Institute should be encouraged to set a standard for other character set processing in China by incorporating these sets using the pattern files used for Chinese in this project. (See item 3 in the following pages.)

DETAILS

1. LECTURES

Twelve lectures were given over a period of twelve days, each lecture occupying a full half day. Twenty students participated in the lectures, with 75% of them coming from the Institute itself, and the other 25% coming from outside the Institute. Of this 25%, some were from MINISIS installations and some were not. For example, one attendee came from the University of Peking.

The lectures were very productive. In addition to formal presentations by the consultants, a question-ansuwer scheme was used in order to encourage the students to develop some of the key ideas themselves. This technique worked extremely well. In addition, a half hour review was conducted before the commencement of each session, in order to ensure that the stuents had grasped the previous days material.

A take-home test was given at the end of the lecture series. The results ranged from 50% to 84%, with the bulk of the class coming in the 70% - 80% range. The purpose of the test was to determine how well the participants had absorbed the material. In marking the tests, the consultants made detailed comments of the work and gave recommendations on how particular questions could be better handled. These recommendations were later discussed with the students.

The participants reaction to this mode of instruction was both positive and very enthusiastic.

A copy of the exam is attached, as well as a copy of the outline for the lecture series.

2. WORKSHOPS

The workshops were held as half-day sessions over a period of seventeen days. All the students (ten) were also participating in the lecture series. Of the workshop students, nine were from the Institute and the other was from another MINISIS installlation in Beijing.

A technique similar to that used for the lecture series was used in the workshop. In formal presentations, the participants were exposed to details of MINISIS internal routines necessary for the creation of new programs. They were instructed in the techniques of structured programming,and through question and answer sessions derived the structured notions of stepwise refinement, were instructed in the use of the SPL programming language, and the use of pseudo-code in program development. The were given several assignments over the period, including hands-on computer assingments. In addition, they were exposed to designs for processors that are of particular importance to the Institute - SDI and Circulation.

The majority of the students performed very well. Topics covered by the workshop are contained in the attachment.

3. ENHANCEMENTS TO MINISIS TO SUPPORT CHINESE CHARACTER PROCESSING

The consultants worked closely with staff members from the Institute (who were also participating in the lecture/workshops) to complete an enhancement to MINISIS for the support of Chinese character processing. The original plan clled for building an interface between the HPCIOS routines and MINISIS; however, this plan was modified in the light of the MINISIS operating context. The HPCIOS routines are not used directly; they were modified somewhat and incorporated directly into new software called 'Handlers' that interface directly with MINISIS. Only the HPCIOS pattern files for the Chinese characters in telegraph code are used directly. Unfortunately, current limitations in MINISIS meant that the technique used to store Chinese characters had to be limited to 10,816 characters. (The original scheme implemented support for over 40,000 Chinese characters; problems arose and this scheme was shelved. IDRC should be encouraged to remove the limitations in order that the extended character set be implemented.)

This enhancement is fully operational and very practical. Demonstrations have been given exhibiting its operation fully. More detail on the work is described in all the attachments.

It is to be noted that the MINISIS approach to alternate character set support is very flexible. Because it uses the software technique of 'handlers', it involves a minimum amount of modification to the programs when new hardware devices are introduced into the system. The Institute will be able to integrate new Chinese character processing hardware into their configuration at their discretion.

4. INTERFACE TO THE PDP/11 SYSTEM

The Scientific and Technical Information Institute of the Ministry of Machine Building in Beijing is currently using two small PDP/11 systems for word processing. This represents a total of four terminals and one printer, one hard disk and two floppy disk units. Data entry clerks have been trained to read Chinese character text and to key in directly the 'phonetic' (not to be confused with Pin-yin) equivalent without using conversion tables, etc. as intermediaries. The system echoes the Chinese character in graphics form as its phonetic is keyed in, so that the operator is aware whether she has made any errors in entry. She can compare the echo to the text from which she is working for validation.

The Institute wishes to transfer this data to the main HP3000 computer, and to have access to it as a data base. However, it has had no means at all of interfacing the PDP/11 to the HP3000. There is no communication hardware and if there were, considerable programming would likely be necessary. The HP3000 has no floppy disk device for reading 8-1/2 inch diskettes; if it did the format would be incompatible.

However, with the use of their HP110 portable microcomputer, Daneliuk and Page were able to devise a method, and test this method, for transferring data between the two machines. They were also able to determine that the PDP/11 based system is using the GB code internally, rather than the telegraph code that the Institute is using with MINISIS.

The HP110 was connected using the existing PDP/11 RS232C cable, to the PDP/11 (the cable was removed from the VT100 terminal, and connected to the HP110). The PDP/11 treated it as a terminal; however, data could be downloaded to the HP110 micro diskette from the PDP/11 data files. The HP110 was then connected as a terminal to the HP3000. The data was copied from the HP110 diskette to an HP3000 file using the FCOPY utility.

As the Institute will be receiving, in the very near future, some HP150 microcomputers, they will be able to use the HP150 to perform the transfer, using the same technique. Institute staff will have to write a conversion program to convert from the internal GB code to the telegraph code in order that the data transferred from th PDP/11 can be used with the MINISIS data base system. More details of the work are contained in the attachments, under the same title.

5. DRAGON TERMINAL

At the time of the consultants arrival, the Institute had just acquired a Dragon terminal. This terminal has a limited set of Chinese characters (6,000) that can be used for both data entry and display. The staff at the Institute were having difficulty in interfacing the terminal to the HP3000; the consultants were able to assist in configuring the terminal and in bringing it to operational status (though not with MINISIS). However, it is now available to the Institute for testing. The terminal apparently uses GB code for representing the Chinese characters so an additional Handler will have to be written to interface it with MINISIS. More information is contained in the attachments under Handler documentation.

6. ALTERNATE CHARACTER SETS

The HPCIOS telegraph code pattern file has been extended to a 5-digit code providing new character patterns representing Cyrillic characters and Japanese Katakana characters, in addition to several other special characters. These could be coded internally in MINISIS as though they were normal Chinese characters. However, this not desirable, the reason being that this code would be completely arbitrary and not standard. Instead these characters should be coded using MINISIS alternate character set conventions. Since both of these character sets are small they can be easily accommodated. IDRC already uses such a standard code for Cyrillic and Arabic.

Nonetheless, HPCIOS pattern files can be used to display these alternate characters, using equipment now available at the Institute. What is required is a minimal extension of the handlers.

The main advantages of this approach are that:

1) a more compact code would be used to represent a character in these sets since they are small - rather than the larger code that would be required using the Chinese character code.
2) the standard coding of these alternate characters will permit the exchange of data with other institutions with minimum or no conversion being necessary.

More details on this topic are contained in the attachments under the same heading.

USING MINISIS WITH CHINESE CHARACTERS

Under a UNIDO contract, Faye Daneliuk and William Page spent one month in Beijing assisting the Scientific and Technical Information Institute of the Ministry of Machine Building by providing training (in the form of lectures and practical hands-on experience) to personnel in data base management systems and programming. Part of the training involved the development of a Chinese language interface to the MINISIS system. This development was carried out by personnel from the Institute, with assistance from Daneliuk and Page. The development involved the programming of special handlers for computer input and output devices used by the Institute; these handlers interface with MINISIS and make possible and practical, the processing of Chinese text with MINISIS.

The work is fully documented - this set of documentation provides guidelines for the MINISIS user for using Chinese with MINISIS.

PART I - HARDWARE

2648A Graphics Terminal:

A handler (Handler13) was written for this terminal. This device has the capability of actually displaying the Chinese characters in their standard graphic form. The user enters the "phonetic" code (not to be confused with pin-yin) and the Chinese character is reflected on the screen.

As soon as the user logs on to MINISIS (on the 2648), the handler executes the terminal set-up. This involves the following.

1. Disengage MEMORY LOCK.
2. Clear the display.
3. Clear the graphics display.
4. Position the cursor at line 18 of the screen.
5. Engage MEMORY LOCK. This reserves the top portion of the screen for Chinese character display (graphics).
6. Set graphic text size to 2.

This set-up results in a screen that will operate as follows:

```
 ______________________________________________
|                                              |
|                                              |
|  Graphics - for Chinese display. 18 lines.   |
| This will display 12 lines of Chinese as each|
| Chinese character is 1-1/2 lines in height.  |
| Scrolls as follows - first 12 lines displayed|
| then display starts at the top again and over|
| writes previous display. The cursor for this |
| segment of the screen is a horizontal line   |
| is present immediately below the line that is|
| currently being displayed.                   |
| The ROLL keys will not function with this part|
| of the screen.                               |
|                                              |
|                                              |
|                                              |
|----------------------------------------------|
|                                              |
| Roman display - remainder of screen (6 lines)|
| Anything entered is reflected on this screen;|
| Chinese characters appear in their numeric   |
| telegraph form.                              |
| This screen operates as a normal HP terminal |
| screen with standard scrolling - the ROLL keys|
| will function with this part of the screen.  |
|______________________________________________|
```

2640A terminals:

A handler (Handler16) was written for this terminal to facilitate input. As the terminal has no graphics mode, output is restricted. These terminals continue to operate in their normal mode, as before. The only difference is that any data stored in Chinese will be displayed on this terminal in the internal telegraph code rather than Chinese. Data entry can be conducted on this terminal in Chinese using 'phonetic code'. See the section on data entry.

2635A terminals:

These terminals have no graphics capability and no handler was written for them as they are not commonly used for data entry. However, staff at the Institute now have the knowledge required to enable them to write a handler if need be. This handler would be virtually identical to that for the 2640.

DRAGON terminal:

No handler was developed during this time for the DRAGON terminal. Institute staff (under the guidance of Wu Kang) are capable of doing this work by themselves. Note that there are serious character limitations on the DRAGON as it currently has a restricted character set.

2608 System Printers:

A handler (Handler15) was written for these devices, as they are capable of graphics output. These printers can output any data stored in Chinese on MINISIS. They can also display the internal 2-byte codes for debugging.

PART II - MINISIS Configuration

The MINISIS Application Programmers Guide and the MINISIS Database Managers Guide contain detailed information on the background, requirements, and use of alternate character sets in MINISIS. As of February 1985, a number of MINISIS internals (among them EXTRACT and DISPLAY) have limitations associated with alternate character set processing - these limitations were discovered during the course of this work. "Work arounds" have been devised. The problems caused by these limitations are identified as they come up in this documentation.

The following files are required in order to run MINISIS properly Chinese. It is the data base manager's (DBM) responsibility to ensure that these are in place.

TERMHDLR.PUB:

This file is an EDITOR file that contains the list of names of the handlers (programs) for the I/O devices that can be used for Chinese character processing. In our case the handlers are HANDLER13 (2648A), HANDLER15 (2608), and HANDLER16 (2640A). Note that any device handler not on this list is handled directly by MINISIS (and therefore will not support Chinese characters).

MESSnn.PUB:

This is the MINISIS message file that can be tailored to an installation. MESS00 refers to the message file as supplied by IDRC with MINISIS. MESS05 is the message file used at the Institute for Chinese character processing. Some of the dialogue contained in MESS05 has been converted to Chinese using MINEDIT. It can all be converted at the discretion of the Institute.

ERRnn.PUB:

This is the MINISIS error file that is supplied by IDRC with MINISIS. It is called in its original form, ERR00. The error messages contained therein can also be converted to Chinese using MINEDIT if desired. The converted file should be called ERR05.PUB.

SYNnn.PUB:

This is the trickiest file of all. It contains the syntax tables for all the MINISIS processors. Under certain conditions (documented in the manuals mentioned at the beginning of this section) it must be modified. Currently, the Institute is using the syntax file as provided by IDRC, called SYN00.PUB. If it is altered for any reason at the Institute, the altered form must be called SYN05.PUB. Great care and caution must be exercised in the modification of this file!!!!!!

UDC files:

Ideally, the DBM should establish UDC (user defined commands) files to make life easier for the users. The UDC command, MINC, is a good example of the type of command that would be useful.

```
MINC
FILE MESS00.PUB=MESS05.PUB.DC
FILE ERR00.PUB=ERR00.PUB.DC
FILE TERMHDLR.PUB=TERMHDLR.HPCIOS.DC
FILE SYN00.PUB=SYN00.PUB.MINISIS
RUN MINISIS.PUB.MINISIS;LIB=P
****
```

Read the MPE manuals to learn more about the creation, assignment, and invocation of UDC's.

Note that under certain conditions (for example, tracking down bugs), it may be desireable to use the printer to display the telegraph codes in their internal form, rather than the Chinese characters in their graphic form. In order to accomplish this, the file equation for the handler must be turned off - do this by entering, at MPE level, RESET TERMHDLR.PUB. Be advised, however, that this disengages all the handlers for your particular session; if you are working on a 2640A it will no longer convert phonetic input to Chinese although it will continue to display the internal 2-byte form of the telegraph code for anything entered in Chinese previously. If you are working on the 2648A it will no longer convert phonetic input to Chinese, nor will it display Chinese characters. Rather, it will display the internal telelgraph codes of anything entered previously. As noted, this mode should only be used for debugging.

PART III - WORKING WITH THE CHINESE LANGUAGE

A code called "Phonetic", not to be confused with Pin-yin, has been developed to facilitate the entry of Chinese characters. This code is based on both radicals and sound - details of the code can be obtained from the Institute. Suffice to say that the Institute already uses this data entry technique for text processing applications on a PDP/11 system (with 4 data entry terminals), and their data entry personnel are highly skillful at reading text in Chinese characters and entering the same using the "Phonetic" code directly (without looking at conversion tables, etc.). In fact, it is probably safe to say that they are almost as fast as data entry clerks working in the Roman languages. It is practical, therefore, to use this code for data entry purposes with MINISIS.

One Chinese character is represented by a code consisting of four Roman alphabetic characters in upper case, for example, WIIR. With MINISIS, a string of Chinese characters would be entered as:

```
\code,code,.....,*  if no Roman characters follow the Chinese
```

and

```
\code,code,.....,*\ if Roman characters do follow the Chinese
```

The backslash (\), the comma (,), and the asterisk (*) are essential for the successful entry of the data. As mentioned, "code" is a four (4) character string of upper case Roman characters.

The handlers convert this code internally to a compact form of the telegraph code, which consists of two bytes (note that the text processing system on the PDP/11 uses the 2-byte GB code for internal representation of the Chinese characters. The compact telegraph code and the GB code are not easily converted, one to another). It is possible, when using MINISIS, to enter the 4-digit numeric telegraph code directly. The form to be used for entry is as follows:

```
\nnnn;nnnn;.....;*  if the Chinese is not followed by Roman
```

and

```
\nnnn;nnnn,.....;*\ if the Chinese is followed by Roman
```

Note the difference in the input form from the "Phonetic" - four digits (combinations of digits from 0 to 9) are entered instead of alphabetic characters; the semi-colon rather than the comma is used as the character delimiter. As before, the backslash and the asterisk are essential.

As well, the 4-digit numeric telegraph code and the 4-character phonetic code can be mixed on input, so long as the appropriate delimiter is inserted after its code (the comma follows the phonetic and the semi-colon follows the telegraph). For example,

```
\nnnn;code,code,nnnn;nnnn;code,nnnn;*
```

and

```
\nnnn;code,code,nnnn;nnnn;code,nnnn;*\Roman string
```

The handlers written for input convert the codes to an internal format. The handlers written for output convert the internal form to Chinese for those devices that can display Chinese characters, and to telegraph codes for those that cannot.

PART IV - USING THE MINISIS PPOCESSORS

Due to existing limitations in MINISIS internal routines, users may encounter some problems when working in the Chinese language under MINISIS. Any processors not mentioned in this documentation were not tested when the documentation was written. The testing of those processors will be carried out by Institute staff at a time convenient to them. Note also that in all cases, for all processors, when a Chinese string is entered at the 2648A terminal in either phonetic or telegraph code, it will appear on the screen in its Chinese graphics form. When entered at the 2640A terminal it will appear in telegraph code format.

1. Any processor that sends output to the line printers will under some circumstances, be off by one or more characters. This most commonly occurs when printing Chinese characters in columnar format (PRINT, LISTFORMAT, LISTDOT, INVERT). There are two reasons for this. First, MINISIS does not consider the blank character (the space) as a Roman character when it is filling blanks between columns. Secondly, under certain conditions (not determined at this time), the MINISIS DISPLAY intrinsic seems to be counting the character set indicators of either the Chinese characters or the Roman characters, but never both at the same time. It should not count them at all when it is filling columns. Workarounds are possible in certain cases in PRINT only, and are described in that section.

2. Another limitation of MINISIS is due to its dependence on the collating sequence of the computer for alternate character sets that are coded as more than one byte. If the system is left to sort the retrieval keys that have been entered in Chinese characters or, to sort data fields for sequenced lists, the resulting order will be that of the numeric telegraph code. This sequence is meaningless for operation with Chinese characters. A special exit has been written for the INDEX processor that will sort Chinese character data on the basis of the phonetic sound position (Pin-yin) sequence. This exit is called CHISORT and should be invoked any time INDEX is being used with a Chinese field for reporting purposes.

Fast access fields that contain Chinese characters cannot take advantage of this sorting procedure even if batch inversion is used. The reason for this is that the B-tree and the KSAM routines themselves use algorithms that are based on the collating sequence of the computer. For the time being the Institute will have to live with this limitation until IDRC extends MINISIS to accommodate this problem. This means that inversion may as well continue to be on-line.

A possible "workaround" to this limitation would be to use a thesaurus of descriptors which would be phrase oriented, and/or a dictionary. The dictionary would consist of: i) Chinese characters in internal code, ii) the character's telegraph number in external representation, iii) the phonetic code for the character, and iv)the Pin-yin word for the character.

3. Another aspect of MINISIS is the extraction rule for inversion (fast access). MINISIS allows for several possibilities - term, word, phrase, and whole field. Whole field and term pose no problem - these have been used successfully in our tests. However, the notions of word and phrase

are different between Roman and Chinese. For this project, it was decided to defer any work on the phrase extraction, and to concentrate on the notion of word extraction. Word extraction is defined to be one Chinese character (two bytes of internal representaticn). A version of USER'GENKEY has been written to perform this type of extraction. It has been successfully tested on the title field in the test MECENG data base; it can be used with other fields as well.

4. MINISIS currently conducts its scans and searches of character strings internally on the basis of bytes, i.e., the alignment is on bytes and not necessarily on even byte boundaries. It has no concept of character size. This means that it is possible for MINISIS to "split" a Chinese character thus leading to a false match or non-match. The use of the internal two-byte code by the handlers minimizes this possibility. If a four byte internal code had been used for the Chinese character, then it would have been more likely that the character, consisting of four bytes, might be "split" more often. Using a four byte code also would make it impossible to print in columns using the MINISIS PRINT processor. Test showed that MINISIS treated the 4 byte internal code as 4 external bytes when, in fact, it was only 2 external bytes. Using the 2 byte internal code has alleviated this problem.

5. Another serious problem in MINISIS is that the EXTRACT routine does not appear to recognize the Chinese character set indicator. The full 2 byte internal code made use of all but 64 characters in the character set. In QUERY mode, MINISIS would ignore the character set indicator and attempt to process the Chinese characters as QUERY operators. This led to unpredictable performance of both QUERY and the QUERY portion of MODIFY. The full 2 byte internal code gave a possibility of over 40,000 Chinese characters. Because of the problem described here, a restricted form of the 2 byte code was implemented. This code allows for only 10,800 Chinese characters - a significant limitation. However, it does mean that QUERY now functions successfully. In the future, when IDRC corrects this internal problem, the data base can be converted to the full 2 byte code.

ENTRY Processor:

The ENTRY processor works successfully with the Chinese character set and mixed Chinese/Roman character set. All ENTRY functions behave as normal, including validation and on-line inversion. Note of course that the inversion is subject to the limitations mentioned at the beginning of this section. Depending on how the data base was defined, the prompting will be in either Chinese, Roman, or both. To enter Chinese characters, follow the instructions given in PART III. Data in a field may be in either character set or mixed. Data entry in Chinese currently may be carried out on the 2640A terminals or the 2648A terminal. Using the 2648A terminal will result in the Chinese data being shown in its graphics form on the screen as it is entered.

MODIFY Processor:

This processor operates successfully with Chinese characters, subject to the limitations mentioned at the beginning of this section. Because MINISIS currently scans strings internally independent of the alignment (byte, word) of the characters, trying to change a single Chinese character in MODIFY may lead to an unpredictable result. Therefore, the user is advised to specify at least two characters when making a change to a Chinese string. Specifying four characters as part of the change string (even though only one needs changing) will ensure that the user does not have strange results. Note also that the retrieval part of MODIFY also works successfully. Enter the Chinese strings as described in PART III. On the 2648A terminal, they will display as true Chinese characters.

PRINT Processor:

Subject to the limitations mentioned at the beginning of this section, this processor operates with acceptable success. The following rules must be observed.

1. When printing a field of Chinese data, ALWAYS reply 0 (zero) to the field prompt, "Number of spaces to the right". To provide spaces to the right of a field containing Chinese data, define a post-literal consisting of at least one blank space, for that field. There should be no suppression specified for the literal. A Chinese character pre-literal can also be specified for a field. The literal is entered following the rules of PART III. Be sure to place at least one blank space or punctuation after the string as part of the pre-literal. Remember that the space is a Roman character. For example, the literal would be entered as shown in the following string:

```
\WIIR,VIIB,*\space or punctuation
```

2. In page set-up, when specifying lines per page, enter one half (1/2) the number of lines that you want. For example, if you wish to print 60 lines per page, you enter 30. The reason for this is the height of the Chinese characters, each character is 2 lines in height. Follow this rule and you won't get into trouble.

3. In page set-up, if you are printing in columns, at this point in time you cannot print more than two columns on a page. If you attempt to set up more than two columns, you will have unpredictable results. Furthermore, when setting up your page in columns, the total page width cannot be greater than 57 Chinese characters (to MINISIS you say 114 in response to page width). Your data will print successfully in columns with minor (not too noticeable) abberations, depending on placement of blanks by MINISIS. The attached format, TPAGECOL, gives an example of an acceptable 2 column print.

Schematically we have

```
<---------------------------------------------->
     Maximum page width 114 (57 Chinese char)
<--------------------->     <------------------->
 Column 1               |      Column 2
                        |
                        |
                        |
                        v
                 Space between columns

  Maximum page width <= Column 1 + Column 2 + Space between columns
```

4. Printing many fields across the page poses no problems.

5. Specifying column positioning for fields (as opposed to page set-up) also poses no problems but is subject to the limitations mentioned previously. The Chinese data wraps around correctly, but slight misalignment can occur because of MINISIS's insertion of and treatment of spaces. The attached format TFLDCOL gives a good example of column printing.

6. Do not try to use field column positioning within a page column set up. The results will be unpredictable in this version of MINISIS.

7. When printing columns (either in page set-up or field set-up, be sure to specify an even number of characters as column width. If the column wide is not even, a two byte Chinese character might be split in half when the line is wrapped.

8. There is absolutely no problem with printing mixed data.

INDEX Processor:

No problems - use the special exit CHISORT for sorting the Chinese characters correctly.

LISTFORMAT Processor:

No problems - subject to the limitations listed above.

INVERT Processor:

No problems, subject to the limitations mentioned previously.

BATCHIN Processor:

No problems. However, conversion to the 2-byte MOD 104 code must be done prior to running BATCHIN as this processor does not use the handlers because it is outside the normal MINISIS system.

LISTDDT Processor:

No problems - subject to the limitations listed above.

SYNCOMP Processor:

No problems.

MINEDIT Processor:

No problems. Use this for modifying the Message and Error files to contain Chinese strings.

QUERY Processor:

No problems now that the restricted two byte code has been implemented. Remember however that the QUERY operators are Roman characters. When Chinese strings are being used in search expressions, if they are followed by operators, you must "shift out" of Chinese. The rules are those described in PART III. The following QUERY statements give you some examples.

```
   i)   =\WIIR,VIIB,*

   ii)  \WIIR,VIIB,*\ OR \KTNW,G6DC,NTIS,*

   iii) TEXT M050 = "\KTNW,G6DC,NTIS,*\"

   iv)  \WIIR,VIIB,*\ * (DANELIUK + \KTNW,G6DC,NTIS,*\)
                      :
                      :
                      :
                      v
                 QUERY 'AND' symbolic form
```

DATADEF Processor:

DATADEF accepts Chinese characters as field names. The only limitation in DATADEF is that the stack size is very large. If the data base manager enters the Chinese string in telegraph code then DATADEF can be run by selecting it from the menu. Currently it is not possible to enter the Chinese string into DATADEF using the phonetic code - this requires a change to the handler. This change will be made by Institute personnel at a convenient time.

In order that the field names used in prompting (ENTRY) and in field displays (ENTRY, MODIFY) be meaningful, at this point in time the Institute will have to run with two RD's for the same physical database, as the PS structure does not allow for the redefinition of field names, only tags. Two RD's can be created by the data base manager. They will have different names, one of the names indicating that that particular data base has Chinese prompting (for example, MECENG and MECENGC). The two RD's will have the same Master, Xref, validation, B-tree, etc. files, and the same field tags and mnemonics. Only the field names will differ -- for one data base the field names will be given in Roman characters, in the other data base they will be given in Chinese characters. The data base with Chinese character field names will be used for ENTRY and MODIFY on the 2648A terminal; the data base with the English field names will be used on the 2640A terminals.

DOCUMENTATION FOR CHINESE CHARACTER HANDLERS IN MINISIS

This documentation describes the handlers written to extend MINISIS to process Chinese characters. Three handlers were written - one for the 2648A terminal, one for the 2640A terminals, and one for the 2608 printers. The work was done as part of a hands-on training project under a UNIDO-sponsered contract.

The two terminal handlers recognize a string of Chinese characters input in either phonetic or telegraph code, and convert the string to an internal two byte per charac er format. This format is based on the numeric telegraph code prefixed by a two byte character set indicator. The internal format allows 8 bits per byte but, because of limitations in internal MINISIS procedures (which result in MINISIS confusing the Chinese string with Roman characters in retrieval commands in both MODIFY and QUERY) , the internal code uses only the upper and lower case Roman alphabet plus the 7th bit (high order) for its code generation - no special characters can be used. This bug was discovered during the first phase of implementation. The original full 2 byte code used all but 34 characters out of the 256 character set. Testing in QUERY demonstrated erratic behaviour on the part of QUERY although the other processors functioned normally (with the notable exception of PRINT which failed for other reasons). A telex from IDRC in Ottawa acknowledged the existence of the problem.

The restricted nature of this version of the 2 byte code means that only 10,816 Chinese characters can currently be supported in MINISIS. The original 2 byte code provided for more than 40,000 Chinese characters. After IDRC corrects the problem, the Institute may choose to revert to the extended 2 byte code - although data conversion would be necessary.

The terminal and printer output handlers recognize the Chinese internal coded form and, for the 2608 and the 2648A, convert it to Chinese graphics. For the 2640A's, the handlers output the telegraph codes of these characters.

The full 2 byte internal code is based on modulo 222:

```
telegraph code N (0-10,815)--->byte 1 = N/222 + 34
                               byte 2 = N MOD 222 + 34
```

The offset 34 in the algorithm avoids all control characters, the space character and the exclamation mark.

The restricted 2 byte code is based on modulo 104. The 104 codes are mapped into 52 upper and lower case Roman characters with bit 7=0, plus 52 upper and lower case Roman characters with bit 7=1.

Note that if the DRAGON terminal is to be used with MINISIS with a character set in the order of 16,000 characters, the problem that was encountered using the modulo 222 algorithm will also arise.

The original design of the handlers called for extensive use of the HPCIOS routines. However, these routines proved to have serious limitations in the MINISIS context and their direct use was abandoned. For instance, HPCIOS is screen-oriented whereas MINISIS assumes a serial input device. However, some source was extracted from them, extended, and then incorporated into the handlers directly. Therefore, none of the handlers use HPCIOS intrinsics directly, and the successful operation of the system is not dependent on them. However, the HPCIOS character pattern file and associated file maintenance programs are used without any changes.

HANDLER13

NAME: HANDLER13 (segment name HANDLER13)

NAME OF SOURCE: HDL13.HPCIOS.DC

PURPOSE: to handle Chinese character input and output on the 2648A terminals

CALLING SEQUENCE: as specified in the MINISIS Application Programmers Guide, section on Alternate Character Sets.

DESCRIPTION OF PARAMETERS: as specified in the MINISIS Application Programmers Guide, section on Alternate Character Sets.

ASSUMPTIONS: the handler assumes a numeric telegraph code based coding scheme for Chinese characters. Each Chinese character is represented by a 2 byte code. The telegraph code, an integer 0 - 10,815 is split into high order and low order parts and stored as two consecutive bytes. High and low order parts are determined by modulo 104 (see previous page). This avoids all non-alphabetic (Roman) characters in the code.

The character \ is used to switch between Roman and Chinese; i.e., it is converted to a %16 %5 or %16 %0, depending on the current character set selected. Input of Chinese characters is by 4 letter 'phonetic' or 4 digit telegraph code. This is translated to telegraph code compacted for internal storage. When Chinese characters are input, the characters are echoed in graphic form on the graphics terminal.

For example: ROMAN\AABC,GGDC,1234;•\ROMAN AGAIN

AABC,GGDC,1234; are codes for Chinese characters.

LIMITATIONS: the use of mod 104 rather than mod 222 for character conversion due to current limitations in MINISIS. The old mod 222 code is still in the source for this procedure - it has just been commented out. Complete validation of the phometic code input is not carried out. An invalid phonetic code produces telegraph code 0000 - equivalent to a blank in the Chinese character string. An incorrect input form, however, such as three letters instead of 4 in a code, etc. is detected.

CALLS: EXITENT
CSERROR

ATTENTION: CSOPEN would normally be called as part of MINISIS'FOPEN but this is not possible in release F.00.00 so the CCBSYS buffer is constructed dynamically by this handler. This seems to be a satisfactory general solution.

Also since MINISIS opens separate terminal files for input and output the FCB file number cannot be used for echoing input (i.e. output) when the handler is called for input.

```
INCLUDE:  TRITRX.HPCIOS.DC     (PB relative arrays for mod 104 code are
                               defined here)
          TYPESINC.SOURCE.MINISIS
          BENDINC.SOURCE.MINISIS
```

EXTERNAL FILES: CSPATN.PUB.CHINESE - character pattern file

```
USES:  Character set code (CHRSET) = 5 -- Chinese
                                   = 0 -- Roman
       Stack address -457  DB minus reserved word: HDLFLG
```

OPERATION: with the 2648 device, a Chinese character has a height of 1 1/2 (one and one half) alpa lines. The 18 alpha lines (22 x 12 = 264 graphics pixels) of graphics screen reserved for Chinese operation will display up to 12 lines of Chinese characters at a time. Since the graphics screen is not capable of scrolling like the alpha screen, the Chinese character portion of the screen wraps around from the bottom to the top of the screen. A long horizontal underscore line indicates the most recent line of text.

The handler checks the HDLFLG word in minus DB to see if the pattern file has been opened. If this word is still zero then the file CSPATN.PUB.CHINESE is opened and its file number is placed in bottom 8 bits of HDLFLG.

The control buffer for the HPCIOS routines (CCBSYS) is then initialized. The terminal open flags in word 0 are set and the file number of the pattern file is stored in word 1.

When the handler is called for input (MODE=1) the terminal is FOPENed for output as STDLIST and the file number is placed in word 2 of CCBSYS. This is to enable the echo of Chinese characters following the input of a Chinese string of codes. This is necessary as MINISIS opens the terminal for input only. In this mode the terminal must be availabe for both input and output. The extra terminal file for output is FCLOSEd before returning to MINISIS and re-opened on the next call for input.

When the handler is called for output, word 2 of CCBSYS is set to the file number provided by MINISIS in the FCB parameter of the handler.

Prior to calling the handler for input, MINISIS has obtained a line of input from the user. This is passed to the handler in the parameters STRING and LENGTH. The handler must process this string to find Chinese character codes and convert these codes to internal form as well as append the Chinese character set indicator code (CSIC %16,%5) to the beginning of Chinese character substrings. The backslash character \ is used to signal a change of character set. If the current mode is Roman (the default) then \ indicates the start of a Chinese character code sequence. If the current mode is Chinese, the \ character indicates return to Roman mode. On return to Roman mode, the CSIC %16,%0 is inserted to mark the beginning of a Roman substring.

The procedure EXITENT is called to perform the translation of Chinese input codes to 4 digit ASCII telegraph code. A return code of 0 from EXITENT indicates succesful translation of all codes in the string. A return code of other than 0 indicates an illegal code or input format. If an error occurs, the handler issues a prompt to re-enter the entire input string. If there is no error, the handler converts the 4 digit ascii telegraph code to 2 byte mod 104 code, described eslewhere in this documentation, as well as to binary form in an integer array. Roman characters are also stored in the integer array using a negative number equal to its ascii value. The integer array is processed by the procedure CSOUT to display the characters (both roman and chinese) on the 2648 graphics terminal.

The 2 byte mod 104 codes with the Chinese CSIC prefix and the Roman codes with the Roman CSIC prefix is passed back to MINISIS in the parameters STRING and LENGTH.

In output mode, the handler converts the internal form (CSICs and mod 104 codes or ascii codes) to binary form in an integer array as described above. Again CSOUT is called to display the Chinese and roman characters on the terminal.

The graphics text mode of the 2648 terminal (text size 2) is used to generate the roman characters. The Chinese characters are generated as 16 by 18 bit patterns. These patterns are obtained from the pattern file based on the numeric telegraph code.

HANDLER15

NAME: HANDLER15 (segment name HANDLER15)

NAME OF SOURCE: HDL!5.HPCIOS.DC

PURPOSE: to handle Chinese character output on the 2608 line printers.

CALLING SEQUENCE: as specified in the MINISIS Application Programmers Guide, section on Alternate Character Sets.

DESCRIPTION OF PARAMETERS: as spec'fied in the MINISIS Application Programmers Guide, section on Alternate Character Sets.

ASSUMPTIONS: the handler assumes a numeric telegraph based coding scheme. Each Chinese character is represented by a 2 byte code. The telegraph code, an integer 0 - 10,815 is split into high order and low order parts and stored as two consecutive bytes. High and low order parts are determined by modulo 104 (see previous page). This avoids all non-alphabetic (Roman) characters in the code.

When Chinese characters are output, the coded characters are printed in graphic form on the line printer. Each Chinese character occupies 2 character positions on the line, with a height of 2 lines. Roman characters are output as 1 character position in 2 lines. They are positioned on the lower of the two lines so they are slightly out of alignment compared to any Roman characters printed on the same line.

LIMITATIONS: the use of mod 104 rather than mod 222 for character conversion due to current limitations in MINISIS. The old mod 222 code is still in the source for this procedure - it has just been commented out.

ATTENTION: CSOPEN would normally be called as part of MINISIS'FOPEN but this is not possible in release F.00.00 so the CCBSYS buffer is constructed dynamically by this handler. This appears to be an acceptable solution.

CALLS: CSERROR

EXTERNAL FILES: CSPATN.PUB.CHINESE

INCLUDE: TRITRX.HPCIOS.DC (PB relative arrays for the MOD 104 code)
BENDINC.SOURCE.MINISIS
TYPESINC.SOURCE.MINISIS

```
USES:  Character set code (CHRSET) = 5 -- Chinese
                                   = 0 -- Roman
       Stack address -457  DB minus reserved word: HDLFLG
```

OPERATION:

This handler only supports output to the line printer. No input mode is provided nor necessary.

The minus DB word HDLFLG is checked to see if the pattern file has already been opened. If the word is still zero, then the pattern file is opened and the file number is stored in the bottom 8 bits of HDLFLG.

The control buffer for HPCIOS is initialized and pattern file number is stored in word 1, the print file number (from the FCB passed by MINISIS) is stored in word 6 and the open flag for the printer is set in word 0.

The string, containing the CSICs, Chinese and roman codes, is converted to an integer array in the format expected by the HPCIOS CSPRINT procedure. Then the CSPRINT procedure is called to display the Chinese and roman characters on the printe. .

Because of limitations in the MINISIS PRINT processor and DISPLAY intrinsic (which cause Roman spaces to be inserted following Chinese strings wthout the appropriate indicator (%16,%0)) , this handler recognizes Roman spaces (blanks) which are not part of the MOD 104 code and treats pairs of Roman blanks within a Chinese string as Chinese single spaces so that they will be processed properly. This is particularly important in the case where the spaces are to the right of the Chinese string.

HANDLER16

NAME: HANDLER16 (segment name HANDLER16)

NAME OF SOURCE: HDL16.HPCIOS.DC

PURPOSE: to handle encoded Chinese character input and output on the 2640A terminals.

CALLING SEQUENCE: as specified in the MINISIS Application Programmers Guide. Section on Alternate Character Sets.

DESCRIPTION OF PARAMETERS: as specified in the MINISIS Application Programmers Guide, section on Alternate Character Sets.

ASSUMPTIONS: the handler assumes a numeric telegraph based coding scheme. Each Chinese character is represented by a 2 byte code. The telegraph code, an integer 0 - 10,815 is split into high order and low order parts and stored as two consecutive bytes. High and low order parts are determined by modulo 104 (see previous page). This avoids all non-alphabetic (Roman) characters in the code.

The character \ is used to switch between Roman and Chinese; i.e., it is converted to a %16 %5 or %16 %0, depending on the current character set selected. Input of Chinese characters is by 4 letter 'phonetic' or 4 digit telegraph code. This is translated to telegraph code compacted for internal storage. When Chinese characters are input, the characters are echoed as telegraph code numbers on this terminal. When Chinese text is displayed on this terminal as output, it is displayed in numeric telegraph form.

LIMITATIONS: the use of mod 104 rather than mod 222 for character conversion due to current limitations in MINISIS. The old mod 222 code is still in the source for this procedure - it has just been commented out. Validation of the phometic code is not carried out. If an illegal code is input it is coded as telegraph code 0000, equivalent to a blank in Chinese characters.

CALLS: EXITENT

EXTERNAL FILES: CSPATN.PUB.CHINESE

INCLUDE: TRITRX.HPCIOS.DC (PB relative arrays for mod 104 code)
BENDINC.SOURCE.MINISIS
TYPESINC.SOURCE.MINISIS

USES: Character set code (CHRSET) = 5 -- Chinese
= 0 -- Roman
Stack address -457 DB minus reserved word: HDLFLG

OPERATION:

This handler is a subset of HANDLER13 for the 2648 terminal. Since the 2640 terminals do not have graphics capability, no display of the Chinese character is possible. The Chinese input codes are, however, processed just as for the 2648 terminal. The Chinese characters are echoed as numeric telegraph codes for verification.

For more details please refer to the HANDLER 13 documentation above.

APPLICABLE INTERNAL TABLES

These are described in the MINISIS Application Programmers Guide in the section on Alternate Character Sets. The tables that were changed in order to operate with the Chinese characters are described below.

CSAT:

This was not modified from the default values provided by MINISIS.

CBT0ST.HPCOS

All the tables have been modified so that no up or down shifting occurs. Also the sort table does not specify any changes in the sorting order. Sorting of Chinese characters in Pinyin spelling sequence is desirable but can not be obtained using a numeric telegraph based code and a 256 entry sort table.

EXITENT

NAME: EXITENT (segment name ENTEXIT)

NAME OF SOURCE: ENT2.KANG1.DC

PURPOSE: to convert an input string of phonetic or telegraph codes (coded Chinese characters) to an output string of telegraph codes.

CALLING SEQUENCE: as it was originally written as an exit, it uses the calling sequence defined for MINISIS user exits as defined in the Application Programmers Guide.

DESCRIPTION OF PARAMETERS:

CNTRL - MINISIS CNTRL record, not used.
DOMDEF - the domain definition, not used.
OWNAREA - work space, not used.
DOMVAL - byte array containing the string to be processed (on input); contains processed string on output.
DOMLENG - integer, contains length of string to be processed; on output contains length of result.
TUPLE - the user's tuple, not used.
CALL'TYPE - integer value, not used.

LOCATION: SL.PUB.MINISIS

ASSUMPTIONS:

This routine assumes that only telegraph codes from 0 to 9999 will be used.

LIMITATIONS:

When this routine actually calls PHOTR to do the telegraph code access when the user has entered her string in phonetic code, the stack grows so large that DATADEF aborts. Thus DATADEF can only be used, at this time, with Chinese strings entered in numeric telegraph code. PHOTR should probably be incorporated into EXITENT, and modified so as not to carry extremely large arrays in memory. This could be accomplished by using KSAM files for the code.

This routine could be modified to return mod 104 code rather than the 4-digit ASCII telegraph code that it returns now. This would mean that it could also be used as an EXIT in MINISIS as it was originally.

CALLS: PHOTR

EXTERNAL FILES: none

INCLUDE: CNTRLINC.SOURCE.MINISIS
BENDINC.SOURCE.MINISIS

OPERATION:

PHOTR is not called if the user has entered the Chinese string in numeric telegraph code. Note that the user can also enter mixed codes. Phonetic codes are terminated with a comma (,), numeric telegraph codes are terminated by a semicolon (;). Also, roman characters can be entered between parenthesis e.g. (ABC) to be translated to their telegraph code equivalent. The telegraph roman characters are treated as if they were Chinese characters and are displayed as large 2 line characters according to a font stored in the HPCIOS pattern file.

PHOTR

NAME: PHOTR (segment name PHO)

NAME OF SOURCE: PHOTR.KANGI.DC

PURPOSE: to locate the numeric telegraph code for a given 4 character phonetic code, GB hexadecimal code, or GB coordinate code for EXITENT. This routine is called once for each 4 character phonetic code encountered in the EXITENT string.

CALLING SEQUENCE:

```
    PROCEDURE PHOTR(ZHICD,TELECD,CHK);
     INTEGER TELECD, CHK;
     BYTE ARRAY ZHICD;
```

DESCRIPTION OF PARAMETERS:

ZHICD - byte array containing the string to be processed.
TELECD - integer, returns to EXITENT the telegraph code if found
CHK - flag to indicate which code table file segment is to be accessed.
CHK = 1 PCOD1.HPCIOS.DC
CHK = 2 PCOD2.HPCIOS.DC
CHK = 3 PCOD3.HPCIOS.DC

LOCATION: SL.PUB.MINISIS

ASSUMPTIONS:

LIMITATIONS:

See the limitations mentioned in documentation for EXITENT. The code segments read in from the files are read into a buffer of 27,001 bytes. This is extremely large, and may impact on the number of users who can use the Chinese facility at one time. Another technique which is more space efficient should be devised - probably the use of KSAM files would be acceptable. The KSAM file would be keyed on the phonetic code. A record in the file would consist of the phonetic code and its telegraph code. This will be relatively easy to implement as it has already been tested.

Another alternative is to implement a TRIE lookup table in PB arrays. The use of PB arrays would allow the table to used simultaneously by many users.

EXTERNAL FILES:

PCOD3.HPCIOS.DC - contains four search intrinsic tables with phonetic code in telegraph sequence.
PCOD2.HPCIOS.DC - contains the GB hexadecimal code.
PCOD1.HPCIOS.DC - contains th GB co-ordinate code.

OPERATION: select the appropriate code depending on value of CHK and use the SEARCH intrinsic to determine the telegraph code for the given input code.

USER'GENKEY

NAME: USER'GENKEY

NAME OF SOURCE: FRSRT1.KANG1.DC

PURPOSE: to carry out "word" extraction for Chinese language processing.

CALLING SEQUENCE: as specified in the Application Programmers Guide section on Alternate Character Set processing.

DESCRIPTION OF PARAMETERS: as specified in the Application Programmers Guide section on Alternate Character Set Processing.

LOCATION: SL.PUB.MINISIS

ASSUMPTIONS: word extraction in the Chinese context is represented by extraction of a single Chinese character (i.e., the notion of word is interpreted as character). This character is, when extracted, represented as 4 bytes - 2 bytes to indicate character set (%16,%5) and 2 bytes of mod 104 code.

LIMITATIONS: when used for on-line inversion the sequence which is used for insertion into the inverted file is the collating sequence of the computer and not the more desireable Pin-yin sequence. This is due to the current limitations of MINISIS.

INCLUDE: BENDINC.SOURCE.MINISIS
TYPESINC.SOURCE.MINISIS

OPERATION: as described in the MINISIS Application Programmers Guide.

CHISORT

NAME: CHISORT

NAME OF SOURCE: CHISORT.HPCIOS.DC

PURPOSE: to generate the correct sorting sequence in INDEX for Chinese string sorting.

CALLING SEQUENCE: as described in the MINISIS Application Programmers Guide under User Exits - INDEX Special Exits.

DESCRIPTION OF PARAMETERS: as described in the MINISIS Application Programmers Guide under the section on User (Special) Exits for INDEX.

LOCATION: SL.PUB.MINISIS

ASSUMPTIONS: the normal ASCII collating sequence does not correctly represent the sequence desireable for Chinese strings. The more acceptable sequence is by sound (or Pin-yin) sequence.

LIMITATIONS: due to MINISIS limitations, this procedure is not available for on-line inversion.

INCLUDE: CNTRLINC.SOURCE.MINISIS
TYPESINC.SOURCE.MINISIS
BENDINC.SOURCE.MINISIS

EXTERNAL FILES:

SRTARY.KANG1.DC - currently, a single record that contains a 9800 word array, in integer format. The telegraph code is used as a subscript into the record; the element value thusly located is the sequential number of its Pin-yin position, in integer.

OPERATION: INDEX passes a string to the procedure. The procedure processes each Chinese character (2 bytes) in the string as follows--first, it extracts the character. Second, it converts the character out of its mod 104 form. Third, it uses this character as a key to read an auxillary SRTARY. Fourth, the Pin-yin numeric position is extracted and replaces the character in the string to be used for sorting. Because it is numeric, it ensures that the Chinese string will now sort in the desired sequence.

For example, telegraph code 0001, which is 'phonetic' YI, is actually 7083 in the Pin-yin sequence.

If the converted character has a value greater than 9799 after step 2, then it is set to integer 0 (zero).

SUPPORT FOR OTHER ALTERNATE CHARACTER SETS, IN ADDITION TO CHINESE, USING HPCIOS PATTERN FILES

The HPCIOS telegraph code has been extended to five digits providing new character patterns representing Cyrillic characters and Japanese Katakana characters, in addition to several other special characters. These could be coded internally in MINISIS as though they were normal Chinese characters, using the 2 byte mod 104 code and the character set indicator %16, %5. This is not desirable, the reason being that this code would be completely arbitrary and not standard. Instead these characters should be coded using MINISIS alternate character set conventions. Cyrillic characters should be preceded by the Cyrillic character set indicator %16,%1; Japanese Katakana by %16,%3. Since both of these character sets are small they can be accommodated as single byte codes. IDRC already uses such a standard code for Cyrillic and Arabic.

Nonetheless, HPCIOS can be used to display these alternate characters, both on the 2648 graphics terminal and on the 2608 line printer. What is required is an extension of the handlers to translate the internal code, i.e., a single byte, into the equivalent numeric telegraph code (a five digit integer). Then HPCIOS-based routines could display them as usual (i.e., the HPCIOS pattern file can be used to display them). For input on non-Cyrillic or non-Katakana terminals (such as the 2640) a method similar to Chinese input (with phonetic codes) could be used such as,for instance, a two character mnemonic. The handlers would have to be extended to recognize a shift to these other alternate characters. Currently a backslash switches between Chinese and Roman; some other character or control key could be used to indicate a shift to one of the other character sets.

The main advantages of this approach are that:

1) only a single byte is used to represent a character in these sets since they are small sets - rather than the 2 bytes that would be required using the Chinese character code. The modulo 104, 2 byte code has only 10,816 characters and is best used to represent only Chinese characters.
2) the standard coding of these alternate characters will permit the exchange of data with other institutions with minimum or no conversion being necessary.

PDP/11 - HP3000 COMPATIBILITY TESTS

The Scientific and Technical Information Institute of the Ministry of Machine Building in Beijing is currently using two small PDP/11 systems for word processing. This represents a total of four terminals and one printer, one hard disk and two floppy disk units. Data entry clerks have been trained to read Chinese character text and to key in directly the 'phonetic' (not to be confused with Pin-yin) equivalent without using conversion tables, etc., as intermediaries. The system echoes the Chinese character in graphics form as its phonetic is keyed in, so that the operator is aware whether she has made any errors in entry. She can compare the echo to the text from which she is working, for validation.

The Institute wishes to transfer this data to the main HP3000 computer, and to have access to it as a data base. However, it has had no means at all of interfacing the PDP/11 to the HP3000. There is no communication hardware and if there were, considerable programming would likely be necessary. The HP3000 has no floppy disk device for reading 8-1/2 inch diskettes; if it did the format would be incompatible.

However, with the use of their HP110 portable microcomputer, Daneliuk and Page were able to devise and successfully test a method, for transferring data between the two machines. They were also able to determine that the PDP/11 based system is using the GB code internally, rather than the telegraph code that the Institute is using with MINISIS.

The HP110 was connected using the existing PDP/11 RS232C cable, to the PDP/11 (the cable was removed from the VT100 terminal, and connected to the HP110). The PDP/11 treated the HP110 as a terminal; however, data could be downloaded to the HP110 micro diskette from the PDP/11 data files. The HP110 was then connected as a terminal to the HP3000. The data was copied from the HP110 diskette to an HP3000 file using the FCOPY utility.

As the Institute will be receiving, in the very near future, some HP150 microcomputers, they will be able to use the HP150 to perform the transfer, using the same technique. The only complication may be the wiring of the HP150 to the PDP/11. The PDP/11 uses only RS232C pins 2, 3, and 7. A null modem cable may have to be prepared to switch pins 2 and 3. This is simple to do. The PDP/11 communicates using 9600 baud, no parity, 8-bit code. The configuration for the HP150 that is required for communicating with the PDP/11 is attached.

Institute staff will also have to write a conversion program to convert from the internal GB code to the telegraph code in order that the data transferred from th PDP/11 can be used with the MINISIS data base system. Note that the GB code is a 2-byte code. After transfer to the HP3000 the order of the 2 bytes has been reversed from the GB order.

CONFIGURATION PARAMETERS

```
DEVICE:  Serial
HANDSHAKE:  None
PARITY:  None
BAUD RATE:  9600
XMIT PACING:  None
RECV PACING:  None
DATA BITS:  8
EOL SEQUENCE:  CR
PARITY CHEC:  On
ECHO:  Off
```

WORKSHOP ON DEVELOPING MINISIS APPLICATIONS - TOPICS

1. When to new develop application programs for MINISIS and when new programs are not necessary.

2. Introduction to Structured Programming with SPL.

   - stepwise refinement
   - psuedo-code
   - BENDINC include file
   - procedures and subroutines

3. Necessary concepts from MPE

   - file access
   - processes
   - data stack and extra data segments

4. MINISIS and the application programmer.

   - the general structure of an application program

5. Data structures available to the applications programmer.

   - user extra data segment

3. Opening and closing the data base.

   - relopen and relclose

4. Reading from, and writing to, the data base.

   - augment and getuple

5. Manipulating data in the data base.

   - domain manipulation routines, add'dom, get'dom etc.

6. Communicating with the user:

   a) Syntax
   b) Dialogue
   c) Error
   d) Parsing
   e) Displaying data

7. Special exits - when, why, and how.

Recommended reading:

Minisis Application Programmer's Guide, pages 1-1 to 1-23, 1-26 to 1-32, 1-58 to 2-60, 4-1 to 4-14.

Attached photocopy entitled, "A User's Interface" from Masters thesis, Daneliuk.

LECTURE SERIES SCHEDULE

1. Data base management systems, Information Management Systems end file systems.

2. Why data base management systems should be the basis of information systems.

3. Classical types of data base management systems

   - hierarchical, network, relational models
   - relational database theory, normal forms

4. The classical implementation of database management systems and information management systems (ORACLE and BASIS)

5. The relational model extended - MINISIS.

   - three schema approach
   - conceptual level, data submodels, etc.
   - internal level, physical structures
   - external level and end-user processors, a functional approach
   - intrinsics, programmer interface

6. Beyond the relational model

   - Attribute Data Model for database design
   - Minisis normal form

TAKE HOME (OPEN BOOK) EXAM

Based on lectures given 11 January - 26 January Beijing

1. A B-tree has the following appearance:

```
                       _________________________________
         Root node---> | Hong Kong   | Singapore   |
                       _________________________________


 ____________________________    _______________________________    ___________
 | Canton   | Edmonton   |       | Los Angeles   | Paris   |        | Turin  |
 ----------------------------    -------------------------------    -----------
```

Assume that each node can contain up to 3 keys.
Draw the appearance of the tree after the following keys have been added, assuming that they are added in the sequence shown. (Use a separate sheet of paper.)

Vancouver
Montreal
Tokyo
Beijing
Shanghai
Fresno
Delhi
Islamabad
Ougadougou
Kathmandu

2. Indicate which of the following statements are true, and which are false, by placing a mark in the appropriate column.

	TRUE	FALSE
a) BASIS is a data base management system.		
b) A data base management system contains an information management system.		
c) Generally, in order to use MINISIS for new applications, it is necessary for a programmer to write new applications		

programs for the user.

d) A good information management system consists of 4 levels.

e) The MINISIS internal level is purely relational.

f) The relational model and the network model both provide a set of mathematically complete operations.

g) A subfield in MINISIS can be repeatable.

h) The power of MINISIS lies in the fact that it supports purely physical data bases.

i) A data base implemented in MINISIS must use KSAM.

j) MINISIS supports only English language data bases.

k) MINISIS can support any charac*er set that can be rigorously defined.

l) In general, a computer system can tell you whether your data is meaningful.

m) If you wanted to implement a personnel system using MINISIS, you would have to write some new programs to support the application.

n) The left outer join, the intersection join, and the union join are all examples of a rooted join.

o) The primary goal of a system is to satisfy end user requirements.

p) The DM in MINISIS is a data submodel.

q) Programmers know what is best for users, in general.

r) MINIsIS is functionally oriented.

s) MINISIS supports variable length data.

t) If a data base is defined internally in MINISIS normal form then, if a field which is defined for the data base does not occur in a recoru, space will be allocated for it in the record.

u) A DS should never be used for code expansion.

v) A fast access file can also be a data base in MINISIS.

w) B-tree stands for binary tree.

x) MINISIS supports only one view of data for an application.

3. Fill in the blanks.

a) ORACLE is an example of a ______________________________.

b) The DATADEF processor is used by the ______________________________.

c) The MINISIS system was designed on the principle of ________ independence.

d) The project list is a characteristic of both ______________________.

e) Within the conceptual level, the RD is closest to the ____________ level.

f) The __________ processor gives the user the capability to retrieve data from a ______________________.

g) The user can use the __________ processor to do special sorts and standard sorts.

h) The _________ processor is __________ independent and can be used to display data in any form on any ________, from a data base.

i) The data base manager is a user at the ______________________ level in MINISIS.

j) The DS and the PS are tools used by the data base manager to define ________________ user ____________, which are commonly called ____________.

k) The hierarchical, the network, and the relational theories represent different data _________.

l) Three characteristics that the conceptual level should have are
____
____
____.

m) If an information system does not have a ______________ level then it becomes data ___________________.

n) List the ouput-oriented processors in MINISIS at the end-user level:
____.

o) The input processors in MINISIS at the end-user level are ______________
____.

p) A ________________ and a __________________________ can be repeatable in MINISIS, but a ______________ cannot.

q) An example of a first normal form relation in MINISIS is a data base that is defined on a ____.

r) Flattening can be used to generate a ______________________________ from a ____.

s) A hit file that is generated in QUERY by a user can be passed to the __________, __________________ and the ___________________ processors for further processing by the user.

t) _________________________ is a example of the use of the retrieval function in the ENTRY processor.

u) An inverted file is called inverted because _________________________
____
____
____
____.

v) A B-tree is an example of an _________________________________.

w) In MINISIS QUERY processor, it (is/is not) necessary for the user to know the fast access paths in order to conduct a search on a data base.

x) _________________________ can be used to implement fast access paths.

y) An RD that is based on a KSAM file can be used simultaneously as a __________________, ____________________, _________________________.

4. Describe the implementation of the following application using MINISIS. Describe the data bases from the user point of view and the data base managers point of view.

The application is to control the flow and allocation of parts and materials in a machine assembly plant. There should be a way to record the components needed to assemble a given machine as well as descriptions of parts. Parts may have components and machines may be components of other machines. The following questions are illustrative of the kind of information that the plant manager would like to obtain from the data base.

1. What parts are necessary to assemble a pump.

2. How many pump handles are currently in storage.

3. In which bin numbers are the parts needed to assemble a hoist located.

4. What are the names and addresses of the suppliers who supply valves.

Feel free to use your imagination to extend this list to indicate what other types of information would be relevant. (In other words, pretend that you are the end user - the plant manager!)