BYPASS2000 for AS/400

User's Guide

Version 3 Release 1 Modification Level 2

SC09-2591-01

BYPASS2000 for AS/400

User's Guide

Version 3 Release 1 Modification Level 2

SC09-2591-01

Note!
Before using this information and the product it supports, be sure to read the general information under “Notices” on page ix.

Edition Notices (September 1998)

This edition applies to Version 3 Release 1 Modification Level 2 of BYPASS2000 (Program 5697–D11) and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure that you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office that serves your locality. Publications are not stocked at the address that is given below.

A form for readers’ comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Canada Ltd. Laboratory
Information Development
2G/345/1150/TOR
1150 Eglinton Avenue East
North York, Ontario, Canada M3C 1H7

You can also send your comments by facsimile (attention: RCF Coordinator), or you can send your comments electronically to IBM. See ″Communicating Your Comments to IBM″ for a description of the methods. This page immediately precedes the Readers’ Comment Form at the back of this publication.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Notices

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications that cover subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Director of Licensing,
Intellectual Property & Licensing
International Business Machines Corporation,
North Castle Drive, MD - NC119
Armonk, New York 10504-1785,
USA.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independent created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Canada Ltd., Department 071, 1150 Eglinton Avenue East, North York, Ontario M3C 1H7, Canada. Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

This publication contains examples of data and reports that are used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses that are used by an actual business enterprise is entirely coincidental.

Trademarks and Service Marks

The following terms are trademarks of the International Business Machines Corporation in the United States or other countries or both:

- Application System 400
- AS/400
- DB2
- COBOL/400
- IBM
- BYPASS2000
- Information Assistant
- OS/400
- RPG/400
- SEARCH2000
- SQL/400
- 400

Other company, product, and service names, which may be denoted by a double asterisk(**), may be trademarks or service marks of others.

About This Guide

This guide describes how to use the IBM BYPASS2000 product on the Application System/400*. BYPASS2000 is a tool that helps you enhance the ability of AS/400 COBOL and RPG applications to handle the coexistence of dates that belong to different centuries.

BYPASS2000 analyzes and rewrites an application to take the coexistence of dates from two different centuries into account. You can choose from several conversion styles to suit the particular needs of your application:

- Fully expanded date fields to accommodate 4-digit years.
- Partially expanded date fields to accommodate 2-digit years with a century flag.
- Windowing logic to interpret dates as belonging either to the 20th or the 21st century.
- Combination of date-field expansion and windowing logic.

Who Should Use This Guide

This document is intended for application developers and programmers who are responsible for analyzing and converting AS/400 applications to accommodate dates that belong to two different centuries.

To use BYPASS2000 you must have AS/400 application and system programming skills, as well as a good knowledge of the programming languages in which your applications are written. You should also be familiar with the details of the application that you plan to convert.

How to Use This Guide

Each phase of BYPASS2000 processing must be performed in a specific order. You must wait for one phase to complete before starting the next one. This is critical for the overall success of the application conversion.

Tip: Before you start processing your application, follow the recommendations in “Chapter 3. Preparing Your Application for Processing” on page 29. This will significantly reduce the number of potential problem situations that you might encounter.

This guide is divided in the following sections:

“Part 1. Introduction to BYPASS2000” on page 1 presents an overview of BYPASS2000 concepts and provides information on preparing your COBOL and RPG applications for processing. It also discusses how to migrate existing conversion environments that you may have created with BYPASS2000 V3R1M1.

“Part 2. BYPASS2000 Tasks” on page 51 guides you through the various phases of how to process your applications using BYPASS2000 and explains menus, displays, and options.

“Part 3. Reference Information” on page 169 contains information on critical aspects of BYPASS2000 processing. It complements the information in the first sections.

The appendixes provide information about the interface files that you can use to import external date-field information into BYPASS2000, and about the markers that BYPASS2000 adds to the converted code.

Conventions Used in This Guide

Commands appear **Like this**.

Coding examples and text that you enter appear `Like this`.

User interface options appear `Like this`.

New terms appear *like this*.

Screen titles, menu names, or keys (for example, the Enter key) have no special appearance.

Obtaining Help

To obtain contextual help for a particular BYPASS2000 display, menu item, option, or parameter, select the element and press F1.

Note: You will not be able to search on the Index in the BYPASS2000 application.

Additional Product Information

For the most up-to-date information that is available about the product, see the BYPASS2000 web site:

`http://www.software.ibm.com/ad/as400/bypass`

This web site is continually updated with the latest product and support information.

You can find technical tips on the AS/400 World Wide Technical Support web site:

`http://www.as400service.ibm.com/as400/service.html`

From this webpage, do the following:

1. Select **Technical Information database** and then select **AS/400 Software Knowledge Base**.
2. Click on **Operating Systems**.
3. Click on **(-) Collapse** to view the topics and then select **YEAR2000**.

Before You Begin

This section contains information that you should know before starting to use BYPASS2000.

System Requirements

In order to use the BYPASS2000 product, you must have the following:

- **Hardware**
  - Application System/400 (excluding Bxx models)
- **Software**
  - Minimum requirements:
    - OS/400 V3R1/V3R6 or higher
    - ADT/400
  - Recommended products:
    - QUERY/400
    - SQL/400
    - Language compilers (RPG/400, COBOL/400)
- **Disk Space**
  - Product requirements:
    - 135 MB (CISC system)
    - 180 MB (RISC system)
  - Conversion environment:
    - 0.3 KB per line of RPG or COBOL code

What BYPASS2000 Supports

The BYPASS2000 V3R1M2 product supports the following source types, languages, file types, and date types.

Source Types

- CBL
- CBL/36
- CBL/38
- CLP (Limited support)
- RPG
- RPG/36
- RPG/38
- RPGLE (Support limited to RPG III migrated to ILE RPG using the CVTRPGSRC command)
- RPT
- RPT36
- SQLCBL
- SQLRPG

Languages

- System/36 compatible COBOL
- System/38 compatible COBOL
- COBOL/400
- CL (Limited support)
- System/36 compatible RPG II
- System/38 compatible RPG III
- RPG/400
- ILE RPG (limited support)

File Types

- Field-reference files (dictionaries)
- Physical and logical files
- External data structures
- Display and printer files
- RPG: RPGLE, RPG36 (but not OCL), RPG38, RPT, RPT36
- COBOL: CBL36, CBL38
- SQL programs: SQLRPG, SQLCBL
- Control Language programs: CLP

Note: BYPASS2000 can convert ILE RPG sources only if they were created from RPG III or RPG/400 using the CVTRPGSRC, and were modified with limited changes.

Date Types

BYPASS2000 adds, removes, or shifts century information only for fields that have been assigned, or reached by propagation, with one of the supported date formats.

Table 1. Date types which are supported in BYPASS2000

Type	Description	Format
001	Year	YY
002	Year at the beginning	YYMMDD
003	Year at the end	DDMMYY or MMDDYY
004	Date with edit mask and year at the end	DD/MM/YY
005	Julian date	YY (sequential day)
006	Month and year	MMYY
007	Year and month	YYMM
010	Century	Cent.
011	Year complement	C-YEA
012	Year, month, day complement	C-YMD
013	Day, month, year complement	C-DMY
015	Year, Julian day complement	C-JUL
016	Month, year complement	C-MY
017	Year, month complement	C-YM
020	Century flag	C-Flag

The date types 011 through 017 correspond to date complements, used in certain types of applications. Date complements are dates represented as integers. For example, the day, month, year complement for August 25, 1997 is 74918002. You determine this value as follows:

```
 99999999
-08251997
--------
=74918002
```

Note: BYPASS2000 is able to identify and automatically assign a date data type field. During the conversion step, BYPASS2000 does not change the date data type. The newly created source will contain the same date data type as the original source.

Using Unsupported Date Types

BYPASS2000 does not support any date type that is not listed in Table 1 on page xiv . If BYPASS2000 encounters a date field in an unsupported format, it highlights the corresponding instruction and requires user information. You must write your own code to add, remove, or shift century information for unsupported date types, and insert it into your application where required.

The BYPASS2000 Software Key

Before you can use the BYPASS2000 product, you must apply the provided software key.

Note: If you do not have a valid software key, see the Program Directory for BYPASS2000 for AS/400, GI10-0406-01 for information about acquiring one.

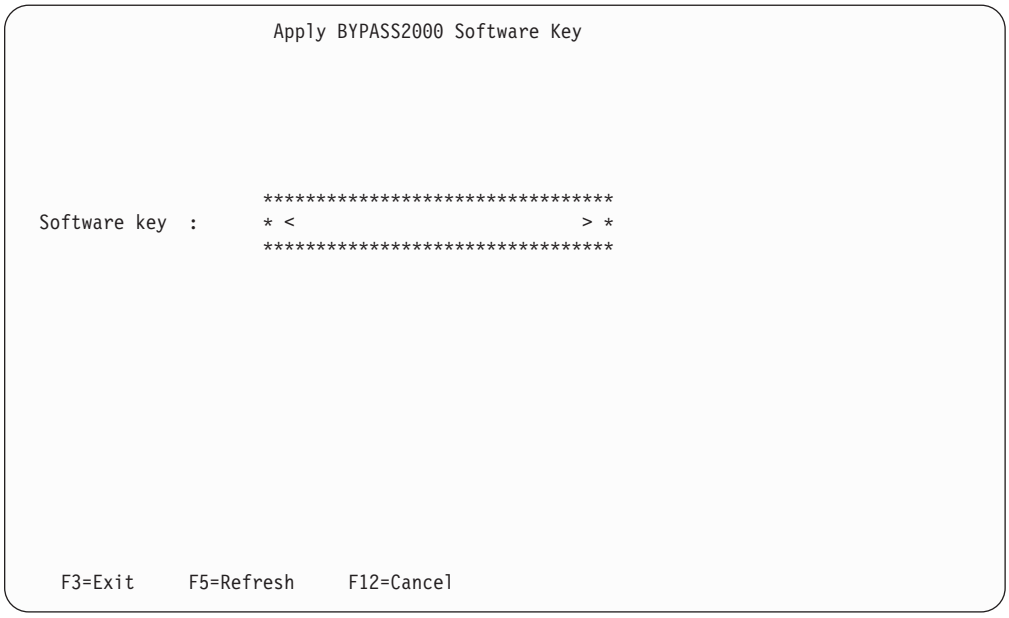
To access the Apply BYPASS2000 Software Key display, perform the following steps:

1. Enter the following commands on an AS/400 command line:
   a. ADDLIBLE QBP2000

      Tip: Skip this step if you already added the BYPASS2000 product library to your library list.

   b. BPPR

      The Apply BYPASS2000 Software Key screen appears.



2. Type the software key number in the entry field and press Enter.

Tip: You can access this display by typing 13 in the BYPASS2000 Environment Setup menu.

Part 1. Introduction to BYPASS2000

Chapter 1. Overview of BYPASS2000

BYPASS2000 is a software re-engineering tool that automatically enhances the ability of AS/400 applications to handle dates that belong to two different centuries. It is primarily an automated code-conversion tool that reduces and simplifies the amount of analysis that is required to identify and convert date fields throughout an entire application.

BYPASS2000 analyzes all application sources and programs. Based on the results of this analysis, it builds a complete storage map of all data used or defined in the application. BYPASS2000 uses this storage map, together with information that it collects during the field assignment phase, to locate and convert all areas of storage that are related to date fields. BYPASS2000 does not use field names to detect date-sensitive fields.

Figure 1 illustrates the chronological order in which you must perform the various BYPASS2000 processing phases.

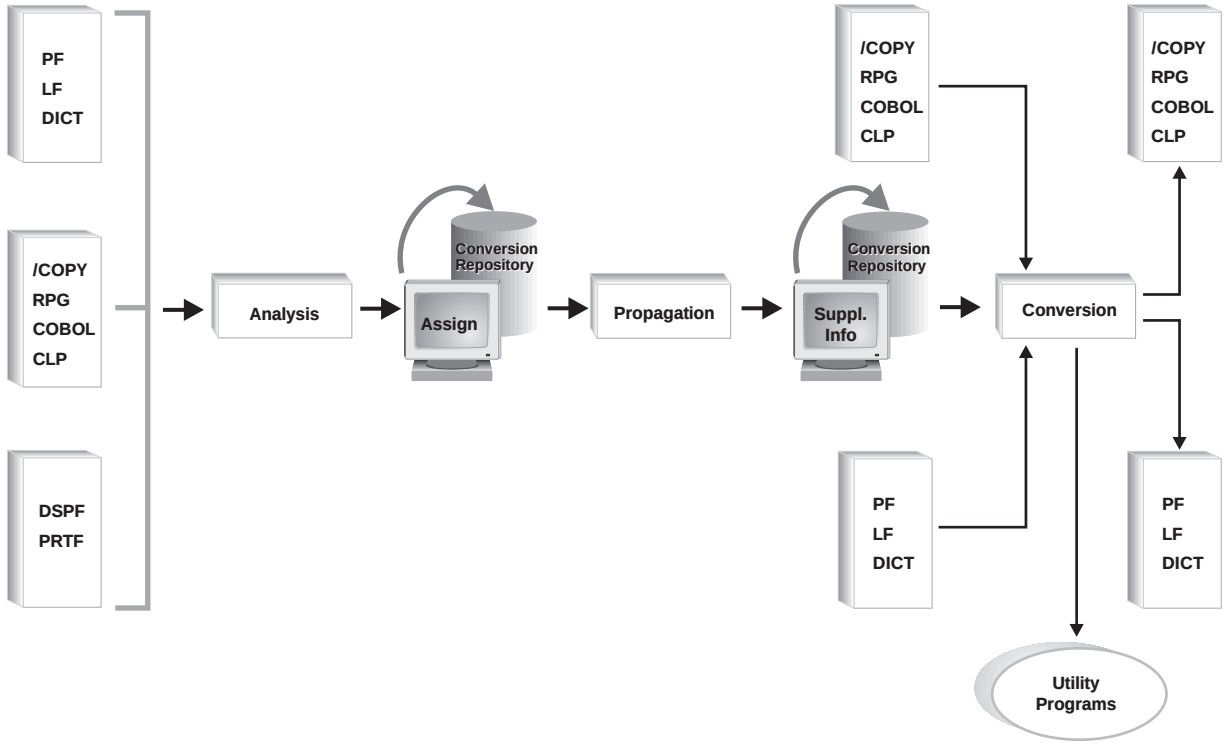


Figure 1. The processing stages of BYPASS2000

The following is a brief description of these processing phases. They are described in more detail in “Chapter 2. Introducing the BYPASS2000 Processing Phases” on page 13 .

Memory-level analysis

BYPASS2000 takes a global view of the application to be converted, and builds an inventory of all parts, such as database definitions, programs, and Copybooks. This inventory is built based on the parts’ storage locations in memory.

BYPASS2000 analyzes the application objects to find data fields and static dependencies between them. It stores the information that describes these fields in a conversion repository that consists of about 80 relational tables.

Note: All components of the application must be available for analysis in order for BYPASS2000 to later recognize all the areas that contain year-related information, as well as all areas that directly or indirectly refer to them.

Date-field assignment

Once BYPASS2000 knows which data areas are part of your application, you must identify all date fields in the application databases and specify their format. This process is called date-field assignment, or ″seeding″. This activity is crucial to the success of the following processing phases, and should be undertaken with the greatest care.

Propagation analysis

From the set of user-defined date fields, and based on the information collected earlier in the conversion repository, BYPASS2000 determines which elements of the application are related to the assigned database fields. It effectively propagates the year attribute from each date-sensitive field to all related fields, throughout the entire application.

Year-related information may exist within dates, in isolation, or mixed with other information, throughout the application. During the propagation analysis, BYPASS2000 updates the conversion repository with information about the elements that need to be expanded to accommodate year information.

At the end of the propagation phase, BYPASS2000 points out any incongruences, such as missing or unreferenced data items. You must resolve these problems before entering the conversion phase.

Application conversion

Once all date-related information has been successfully propagated throughout the application, BYPASS2000 can automatically re-engineer all components of the application. It rewrites sources one at a time, based on the information that is archived in the conversion repository.

BYPASS2000 rewrites the application in the way that it could have been written at the time of the original development, if the coexistence of dates from two different centuries had been taken into account.

BYPASS2000 modifies all physical file fields that have been identified as date fields according to the conversion style that you have chosen for your application and your specifications during the date-field assignment phase. BYPASS2000 may increase the width of all year-related fields found in programs, or of a subset of these fields only. By default, date fields in printer and display files are not expanded.

Typically, BYPASS2000 is able to modify most of the impacted statements in programs and physical files automatically. It highlights the remaining statements which you must then reviewed for manual changes that may be required.

During this phase, BYPASS2000 generates programs, COPYs, and DDS source members that have been modified according to the chosen conversion style.

Creation of utility programs

Once BYPASS2000 has converted your application, it can generate a number of utility programs that you can use to migrate the databases of your application, create test data, or test for valid data in expanded date fields.

Choosing the Right Conversion Style for Your Application

Once you specify a conversion style, it applies to all propagated fields, as well as to the assigned dates with expansion type 0. This allows compatibility with environments that you may have migrated from an earlier BYPASS2000 version.

Note: Any dates assigned in a previous environment set their expansion type to 0.

Each conversion style is briefly described below. Refer to Table 2 on page 6 which shows the result of using a particular conversion style.

Full windowing without date-field expansion

BYPASS2000 adds windowing logic to your code instead of expanding date fields. Unless otherwise specified, date fields keep their original length. Year information greater than the minimum YEAR value will be interpreted as belonging to the 20th century. Year information lesser than the minimum YEAR value will be interpreted as belonging to the 21st century. The default boundary-year value is 50. See “Minimum YEAR value for 20th century” on page 11 for information on changing the default value.

The full windowing technique keeps the number of date fields that need to be expanded to a minimum and simplifies the date propagation phase of the BYPASS2000 processing cycle. It also provides an advantage if you need to convert several programs that interact with each other. Because you do not expand any LINKAGE parameters, you will not experience a conflict between programs that have already been converted and programs that have not.

However, there are some limitations. You cannot apply this technique to dates that are part of a file key. If you select full windowing, you cannot add logical views that contain a field that was not expanded, because the absence of century information would result in incorrect sorting.

Date-field expansion to add a century flag

BYPASS2000 expands date fields by a single digit. This digit represents the century flag and can contain different values to indicate different centuries. Generally, the value 0 indicates that the date belongs to the 20th century, while the value 1 indicates that the date belongs to the 21st century.

This expansion is logically equivalent to the full expansion of a 2-digit year to a 4-digit year. Choosing this conversion style allows your customized application to interface correctly with standard packages that make use of the century flag.

Full date-field expansion

All date fields will be expanded to accommodate 4-digit year information. This applies to 2-digit years as well as to 3-digit years that contain a century flag. You can decide not to expand date fields in display and printer files. In this case, BYPASS2000 will add windowing logic where necessary.

No expansion of date fields that contain a century flag
Date fields that currently contain a century flag will not be expanded, and the century flag is preserved in the converted code.

You can change the conversion of a new or migrated environment without making any changes to the existing field assignment. Simply specify a different conversion style and run the propagation phase again.

Table 2. Date-field results after completing the conversion phase

Conversion Style	Description	Expansion Type	Date Field before Conversion	Date Field after Conversion	Comment
1	Full windowing	not expand	YY	YY	Windowing logic instead of date field expansion
3	Date field expansion to add century flag	expand + one reusing	YY	CYY	Used for packed fields
		expand + one	YY	CYY	Used for all other fields
4	Full date field expansion	expand + two	YY	YYYY	
			CYY	CYYYY	Used for date fields that contain a century flag
		not expand	YY	YY	Used for date fields in printer and display files
		default	YY	YYYY	
5	No expansion of date fields that contain a century flag	default	CYY	CYY	
			YY	YYYY	

See “Choosing the Right Conversion Parameters” on page 8 for information on how to specify the conversion style for your application.

Selecting the Right Expansion Type

Default Conversion style	Expansion type in Seeding	Original Year format C = Century flag i.e 0 CC = Century i.e 19 YY = Year	Converted Year format
All	1	YY	YY
All	1	CYY	CYY
All	1	CCYY	CCYY
All	2, 3	YY	CYY
All	4	YY	CCYY
All	4	CYY	CCYY
1	0	YY	YY

Default Conversion style	Expansion type in Seeding	Original Year format C = Century flag i.e 0 CC = Century i.e 19 YY = Year	Converted Year format
1	0	CYY	CYY
1	0	CCYY	CCYY
2, 3	0	YY	CYY
4	0	YY	CCYY
4	0	CYY	CCYY
5	0	YY	CCYY
5	0	CYY	CYY

Exchanging Data between Date Fields of Different Formats

The BYPASS2000 conversion engine can handle data exchange between fields that contain dates with and without century information or a century flag. The following relationships are acceptable, even if the dates are imbedded in fields which do not contain date information.

Table 3. Exchanging data between date fields of different formats

From Date Format	To Date Format
Year only:	
CCYY	YY
CCYY	CYY
CYY	YY
CYY	CCYY
YY	CCYY
YY	CYY
Year and Month:	
CCYYMM	YYMM
CCYYMM	CYYMM
CYYMM	CCYYMM
CYYMM	YYMM
YYMM	CCYYMM
YYMM	CYYMM
Julian Year:	
CCYYDD	YYDDD
CCYYDDD	CYYDD
CYYDDD	YYDDD
CYYDDD	CCYYDDD
YYDDD	CCYYDDD
YYDDD	CYYDDD
Year, Month, and Day:	
CCYYMMDD	YYMMDD
CCYYMMDD	CYYMMDD

Table 3. Exchanging data between date fields of different formats (continued)

From Date Format	To Date Format
CYYMMDD	YYMMDD
CYYMMDD	CCYYMMDD
YYMMDD	CYYMMDD
YYMMDD	CCYYMMDD
Day, Month, Year:	
DDMMCCYY	DDMMYY
DDMMCCYY	DDMMCYY
DDMMCYY	DDMMYY
DDMMCYY	DDMMCCYY
DDMMYY	DDMMCCYY
DDMMYY	DDMMCYY
Month, Day, Year:	
MMDDCCYY	MMDDYY
MMDDCCYY	MMDDCYY
MMDDCYY	MMDDYY
MMDDCYY	MMDDCCYY
MMDDYY	MMDDCCYY
Month, Day, Year:	
MMDDCCYY	MMDDYY
MMDDCCYY	MMDDCYY
MMDDCYY	MMDDYY
MMDDCYY	MMDDCCYY
MMDDYY	MMDDCCYY
MMDDYY	MMDDCYY

Choosing the Right Conversion Parameters

You can set a number of parameters to customize how BYPASS2000 converts your application. The screen below shows the first panel of the Customize Conversion Parameters display.

```
                    Customize Conversion Parameters


Conversion-database library  . . . . . . . . . . . . . . . . . . : MONAXXDB
System 36 COPY library . . . . . . . . . . . . . . . . . . . . . : *NONE
Default conversion style . . . . . . . . . . . . . . . . . . . . : 4
Column indentation: 1-9. . . . . . . . . . . . . . . . . . . . . : 4

Use caller-called relationships during propagation . . . . . . . : Y
Maximum level of propagation tree  . . . . . . . . . . . . . . . : 012
Maximum elapsed time for analysis of one program . . . . . . . . : 0003600
Maximum elapsed time for propagation of one program  . . . . . . : 0003600
Maximum elapsed time for conversion of one program . . . . . . . : 0003600
Remove REM marker from converted lines . . . . . . . . . . . . . : N
Assume that value of program-name variable was verified  . . . . : N
Insert marker on the right ('Y')instead of the left ('N'). . . . : N
Insert SHIFT instruction on system date  . . . . . . . . . . . . : Y
Generate instruction with quote('Y') instead of apostrophe('N'): Y



                                                                More...

F3=Exit        F5=Refresh        F12=Cancel        F9=Save
```

These parameters are stored in the BPPARM data area of the *xxxxxx*DB library. See “Customizing Conversion Parameters” on page 58 for information on how to specify the conversion parameters.

Conversion-database library

Specify the name xxxxxxDB of the library that is associated with your environment. This library contains the parameters that were set up when you created the conversion environment.

System 36 COPY library

Specify the library to be included for the COPY statements that BYPASS2000 generates.

Default conversion style

Specify the conversion style that you want BYPASS2000 to use for all date fields that you have not assigned, and for the fields that you have assigned with the default expansion type. Valid values are:

- 1 = Full windowing (no field expansion)
- 3 = Expand date fields by 1 digit, adding a century flag
- 4 = Expand date fields, adding 2-digit century information, even if the original field already contains a century flag.
- 5 = Expand date fields by 2 digits if the original code contains 2-digit Year information.

Column indentation

Specify the value for the default column indentation for the new statements that are inserted in the converted sources.

Use caller-called relationships in propagation

Specify whether or not you want to use the relationship that exists between a caller program and its callee, during the propagation phase.

- Specify ’Y’ if you want BYPASS2000 to propagate date information from caller programs to called programs, and vice versa.
- Specify ’N’ if you want BYPASS2000 to ignore the relationship that exists between a caller program and the called program, during the propagation

analysis. In this case, BYPASS2000 assumes that the linkage parameters do not contain date information, or that the dates have been assigned.

Maximum level of propagation tree

The propagation-analysis process consists of a number of iterations. During each iteration, memory areas that contain date data pass the date attribute on to areas with which they share elementary relationships. The process ends when all the elementary relationships involving dates have been processed.

Specify the maximum number of cycles after which the propagation analysis process should be terminated. The default is set to 12 iterations.

If BYPASS2000 encounters a problem, the process will not come to completion in the set number of cycles, and you will receive an error message alerting you to the problem.

Maximum elapsed time for the analysis of a single program

Identify the maximum elapsed time for the analysis of one program.

Maximum elapsed time for the propagation of a single program

Identify the maximum elapsed time for the propagation of a single program.

Maximum elapsed time for the conversion of a single program

Identify the maximum elapsed time for the conversion of a single program.

Remove REM marker from converted lines

This parameter specifies whether you want BYPASS2000 to add the BP2REM marker to your converted RPG and ILE RPG sources. This marker identifies comments that signal the beginning and end of lines of code that BYPASS2000 has added to your source code. It also marks information for highlighted conversion problems.

Assume value of program-name variable was verified

This parameter specifies whether you want to be prompted to verify program names when BYPASS2000 makes assumptions about the names of called programs.

Since BYPASS2000 also propagates date fields that are passed as parameters to other programs, it needs to know the names of called programs. If a program uses a variable to specify the name of the program to call, BYPASS2000 will ascertain the program name from the program logic.

Insert marker on the right instead of the left

This parameter specifies where you want BYPASS2000 to insert markers in the converted RPG and ILE RPG sources.

- Specify 'Y' to insert the markers in columns 76-80 (comment positions)
- Specify 'N' to insert the markers in columns 1-5 (page and line positions)

Insert SHIFT instruction on system date

Specify whether BYPASS2000 must insert a SHIFT instruction for the system date.

- If you specify 'Y', BYPASS2000 includes code in the converted programs to intercept the value of the system date when it is retrieved (using typical language functions such as UDATE or UYEAR in RPG and ACCEPT in COBOL). It also includes code to add a number of years to the actual system date.

This will simulate that the application program is running at a later date in time (for example, 1999 or 2000). If you want your system dates to be shifted forward in time, for testing purposes, you must also specify the shift-value parameter.

- If you specify 'N', the system-date values will be retrieved normally and will **not** be modified by the date shifting logic in the converted application.

The BYPASS2000 date-shifting logic for system dates intercepts only the common language interfaces for system dates. There are other ways to retrieve system dates in applications, such as by parameters passed between programs, or by using system APIs. BYPASS2000 will not be able to intercept all the possible ways of retrieving the system date.

Note: Year shifting is intended to be used only for the period of time when the converted application is being tested. For production use, the application should **not** use date-shifting logic for the system dates.

Generate instruction with quote ('Y') or apostrophe ('N')
For COBOL programs, this parameter specifies whether BYPASS2000 must place constant values between quotes (″) or apostrophes (').

- Specify 'Y' for quotes.
- Specify 'N' for apostrophes.

Flag value for 20th century (19xx)
Specify the value for the century flag that identifies the years between 1900 and 1999. This value is used when you customize environment COPYs.

Flag value for 21st century (20xx)
Specify the value for the century flag that identifies the years between 2000 and 2099. This value is used when you customize environment COPYs.

Shift value
Specify the number of years that you want BYPASS2000 to add to the system dates when it encounters a SHIFT instruction. This value will also be added to database dates by the test-migration programs. It is also used to customize environment COPYs.

Range of constant values used for comparing or setting date fields
This is the range of values for which the century is added to a date field or to a constant (not a variable) used as a date (in comparisons or other operations on dates).

For example, for a range of values from 50 to 99, the 2-digit-year comparison:

```
IF YEAR > 90
```

becomes the 4-digit-year comparison:

```
IF YEAR > 1990
```

However, the 2-digit-year comparison:

```
IF YEAR > 49
```

remains unchanged.

BYPASS2000 highlights any situation that requires manual changes to the source.

Minimum YEAR value for 20th century

This parameter is used for constants. It lets you dynamically apply the ″window″ concept for automatic century addition or removal. You define the boundary-year value below which the year is attributed to the 21st century, and above which it is attributed to the 20th century.

The default value is 50, and has the following impact:

- For any 2-digit year that contains a value between 00 and 50, the century digits are set to ″20″.
- For any 2-digit year that contains a value between 51 and 99, the century digits are set to ″19″.

When you change this parameter to a value other than 50, you must also customize the environment COPYs so that the ″window″ is interpreted correctly when your application is running. See “Customizing Environment COPYs” on page 68 for details.

Maximum number of propagated links for a date (99=all)

Specify the maximum number of propagated links. If a date field is linked to more fields, propagation stops for that program and BYPASS2000 generates an inquiry message. If you specify 99 for this parameter, there is no limit for the number of links.

Stop propagation when a second year is propagated to a 6/7 digit field

Specify ’Y’, if you want BYPASS2000 not to propagate a date to a 6-digit field that already contains a date in a different position. In this case, BYPASS2000 ignores the new date that it has found and issues an inquiry message.

Specify ’N’ if you want propagation to work as usual. In this case, BYPASS2000 propagates a date to a 6-digit field, even if the field already contains a date in a different position.

Create new dictionary (’Y’) or convert existing dictionary (’N’)

Specify ’Y’ if you want to create a new dictionary.

Specify ’N’ if you want to convert the existing dictionary.

Dictionary name

If you have set the previous parameter to ’Y’, you must specify a name for the new dictionary that you want to create.

Dictionary conversion

Specify how you want BYPASS2000 to convert a dictionary. Valid values are:

- 1 specifies that the field definition for expanded date fields should be modified.
- 2 specifies that the field definition for expanded fields should be duplicated and placed after the original field definition.
- 3 specifies that the field definition for expanded fields should be duplicated and placed at the end of the source.

Chapter 2. Introducing the BYPASS2000 Processing Phases

This section describes the BYPASS2000 processing phases, from the setup of the conversion environment to the conversion of your application. The diagram in Figure 2 on page 15 shows the processing steps that correspond to these phases.

Setting up a Conversion Environment

To begin processing an application with BYPASS2000, you must set up a conversion environment. The purpose of the conversion environment is to identify and classify the components of the application that you want to convert:

- Program source files including DDS
- File objects (PF, LF, DSPF, PRTF)
- COPYs containing file descriptions.

The conversion environment contains information about the location of the original files, as well as the location of the modified source code thatBYPASS2000 generates during the application conversion.

The name of the environment is called the conversion identifier. It will be used with different suffixes to name a number of BYPASS2000 libraries. Specify a unique name *xxxxxx* of your choice. This identifier is always used with the suffix DB as the name of the BYPASS2000 repository library, and with various suffixes to create the library names.

Table 4. Library names based on the conversion identifier

Library Name	Library Content	Default Environment	Customized Environment
xxxxxxOLD	Original sources	X	
xxxxxxY2K	Converted sources	X	
xxxxxxDAT	Original FILE type objects	X	
xxxxxxDB	Conversion repository tables	X	X
xxxxxxOBJ	Compiled objects	X	

From a system point of view, the conversion environment represents the library that is used as a repository for all the information that BYPASS2000 gathers during the various processing phases. The name of this library is composed of the conversion identifier, followed by the suffix DB. For example, the environment `MYENV` is represented by the library `MYENVDB`.

Creating a Default or Customized Environment

You can choose between a default or a customized environment. If you choose to set up a default conversion environment, you must load all application source in the appropriate files. You must also load all file objects in the libraries that are specified during the environment creation. If you choose to set up a customized conversion environment, you must specify the libraries that contain the application files and file objects. File objects provide BYPASS2000 with information about database relationships and file structures.

If you create a default environment, BYPASS2000 sets up default libraries for you and creates empty source files in the libraries *xxxx*OLD and *xxxx*Y2K that will contain the original and converted members, as well as the library *xxxx*OBJ that will contain user-created compiled objects. You are responsible for moving the original sources and file objects to these libraries, as described in “Loading Source and File Objects” on page 65.

If you create a customized environment, you can specify your own libraries and source files as the source and target for the subsequent analysis and conversion activities.

Note: If you decide to create a customized environment, you should first make copies of your COBOL source (Copybook and program sources). This is necessary, because during the analysis process, BYPASS2000 writes the line

```
******** BYPASS2000 NORMALIZATION ********
```

to the COBOL source. The presence of this line could create problems if you use the Interactive Source Debug Utility.

Refer to “Chapter 6. Setting up the Conversion Environment” on page 55 for the required steps to set up a conversion environment.

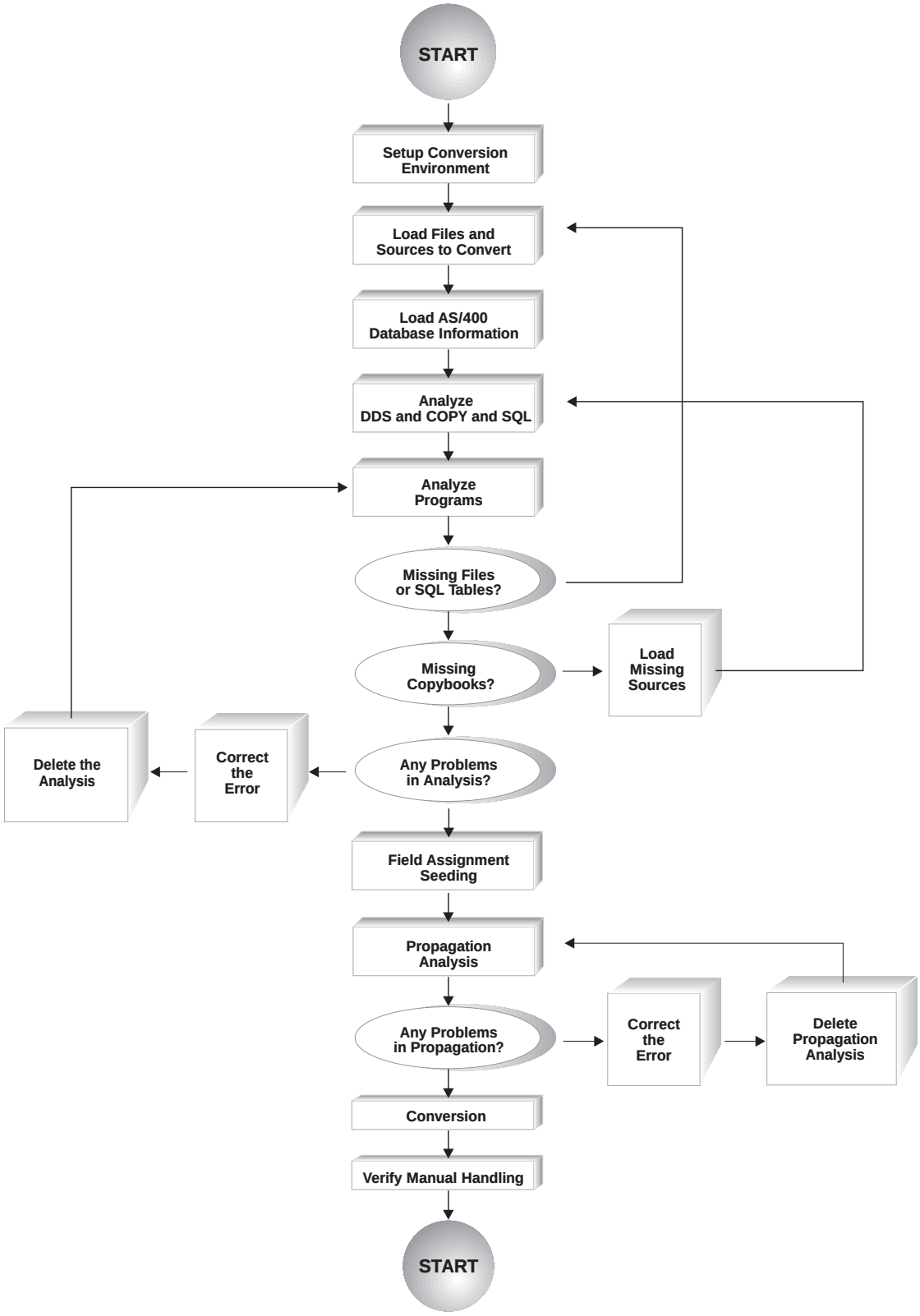


Figure 2. The BYPASS2000 processing flowchart

Loading Information About Application Databases

Once you have created a conversion environment, you must load your application databases. This allows BYPASS2000 to analyze the structure of all the application files in the libraries that were specified during the creation of the conversion environment.

Loading the objects contained in the *xxxx*DAT library provides BYPASS2000 with the following information:

- The structure of the files' record format and their relationship with the database dictionary (field reference file).
- The relationships between physical and logical files.
- The structure of keys and indexes.
- The relationship between objects and sources.

See "Chapter 8. Loading User-Database Information" on page 77 for the steps that are required to load your application's databases.

Memory-Level Analysis

During the memory-level analysis, BYPASS2000 examines all source statements that are related to memory definition and management, and creates a repository of the storage areas (fields described by both databases and programs) of your application. Every field definition is stored in the BYPASS2000 conversion repository and every instruction is stored as a relationship between two or more storage areas.

The memory-level analysis takes place in the following order:

1. Analysis of DDS.
2. Analysis of COPY members (containing file descriptions).
3. Analysis of SQL table-definition sources.
4. Analysis of program sources.

Each analysis step must be completed before you can proceed to the next one.

See "Chapter 9. Analyzing Your Application" on page 79 for the steps that are required to perform the memory-level analysis of your application.

Mapping Data Declarations to Storage Areas

Based on the components of the application to be converted, BYPASS2000 builds a network representation of the application data.

First, BYPASS2000 analyzes all data declarations in the application and maps the data declarations to storage in memory. This creates one node for each storage area.

Next, BYPASS2000 analyzes all program instructions, to determine the relationships between fields. These relationships are represented as links between nodes.

Figure 3 on page 17 illustrates the storage mapping for a piece of source code.

```
I                                1    6    X
I                                7    9    Y
I                                10   12   Z
C          MOVE X     Y
C          MOVE Y     Z
C     X    IFEQ       Z
```

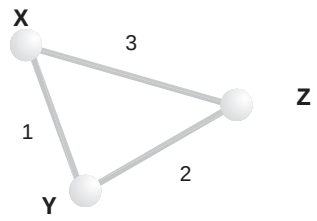


Figure 3. Application data represented as a network of nodes and links

Understanding BYPASS2000 I/O Areas

In many BYPASS2000 menus and displays you will encounter the term *I/O area*. I/O areas are memory locations that correspond to file fields or data structures. BYPASS2000 names I/O areas as follows:

File field
The I/O-area name is the same as the record-format name preceded by an asterisk (*).

Field not contained in a data structure
The I/O-area name is the same as the field name.

Field contained in a named data structure
If a program contains a named data structure, BYPASS2000 uses the name of the data structure as the I/O-area name.

For example,

```
IDATATY          DS
I                                       1   60ZDATF
I                                       1   20ADATF
I                                       3   40MDATF
I                                       5   60JDATF
```

In this case, the data structure DATATY corresponds to the I/O area *DATATY.

Field contained in an unnamed data structure
If a program contains one or more unnamed data structures, BYPASS2000 assigns sequence numbers as a way of differentiating I/O areas that correspond to the different unnamed data structures.

The following example illustrates a file (TEST1) which contains two unnamed data structures.

```
I             DS
I                                       1   60FIL01
I                                       7  100FIL02

I             DS
I                                       1   6 FIL03
I                                       7   8 FIL04
```

When BYPASS2000 analyzes TEST1, it names the I/O areas corresponding to the unnamed data structures as follows:

```
DS-source_name-sequence_number
```

The I/O area corresponding to the first unnamed data structure in TEST1 will be named DS-TEST1-00001. The I/O area corresponding to the second unnamed data structure in TEST1 will be named DS-TEST1-00002.

Date-Field Assignment

Once BYPASS2000 has built a detailed map of all storage areas, you must provide date-related information for all database fields. Identifying which fields are date fields, marking them, and specifying their exact format is called date-field assignment (alternatively anchoring or seeding).

You must identify all date-related fields in databases, and those that belong to the layout of physical files. For each one, you must decide whether to widen it or add windowing logic. This process is crucial to the successful conversion of the application. The more time you spend to ensure that all date fields are assigned correctly, the more satisfactory your conversion results will be.

Typically, you only need to identify date-sensitive fields in the application's databases. BYPASS2000 uses this information, together with its knowledge of language constructs and syntax, to identify all the date-related fields within programs.

It is important to identify the format of the dates, such as DMY or YMD, so that BYPASS2000 can track the year information. (The positions of month and day within a date field are not important.)

Once date fields have been assigned, BYPASS2000 will use them as starting points for the subsequent propagation and conversion activities. If you provide insufficient information about date fields and their formats, the conversion of your application may produce unexpected results.

Note: It is critical to identify *all* database date fields and their formats correctly. For this task, you should have expert knowledge of the application. Any database field that is not assigned as a date field is automatically considered to be a non-date field and therefore will not be converted.

The following list briefly describes how to handle the different types of date-related fields during date-field assignment:

Date fields in database files
: You must assign the date-related fields in databases.

Date fields in COPY
: You do not need to assign date-related data in COPYs that are not associated with a file. If you want to assign them, you can set a lock condition for the entire data structure.

Date fields in programs
: You do not need to assign date-related fields in programs, because BYPASS2000 will find them during the propagation phase. However, assigning one or more internal fields can shorten the propagation process and can clarify whether the area or field is date related, or what its date type should be. This may become necessary after a first run of the propagation analysis, when the propagation tree is too deep or too big for the propagation analysis to terminate properly.

Pre-assigned date fields
: Some date fields are automatically assigned by BYPASS2000; for example, system-date fields, such as UDATE, YEAR, or *DATE.

You can assign date fields manually or you can import date-field assignment from various sources. See “Chapter 11. Assigning Date Fields” on page 99 and “Chapter 12. Importing Date-Field Information” on page 117 for detailed instructions.

Assigning Fields and I/O Areas

The following list shows the various options on the BYPASS2000 Date-Field Assignment menu that you can use to assign fields and I/0 areas.

1. Assign date field for I/O area related to file (mandatory)
Use this option to assign date fields located in database files. You must assign all the date fields that belong to the layout of a physical file. Once these date fields have been assigned, they will be used as starting points for the subsequent propagation analysis and code conversion.

2. Assign date field (optional)
Use this option to assign date fields in COPY and program sources. Assigning date fields in COPY and program sources can speed up the subsequent propagation phase and decrease the number of generic areas that BYPASS2000 may find. Once these date fields have been assigned, they will be used as starting points for the subsequent propagation analysis and code conversion.

3. Assign I/O area to related file (optional)
Use this option to specify that you want a physical-file layout to be described by a COPY source, instead of the DDS source. It is very useful when you want to assign date fields in files that are not externally described. When you link a COPY source with a physical file, BYPASS2000 uses the field layout that is declared in the COPY. This information can help you to seed the file using a detailed field description.

4. Assign record type to related I/O area (mandatory)
Use this option to assign date fields in files that contain multiple record formats. You must specify the position of each variable to identify the record layout. This information will be used to generate the migration programs for multiple-format files.

9. Assign dates from files to program and COPY areas (optional)
Use this option to automatically assign date fields in a file to related program or COPY areas. It helps you quickly assign dates in program and COPY areas that are used to read multiple-format files. You can select each format of the file, and specify the program or COPY area that contains the selected format. BYPASS2000 will assign each date that you assign for this format to the proper field in the selected source area.

Rules for Assigning Date Fields

You must follow certain rules to assign a date field correctly:

- For date types with a year length of 4 you must specify expansion type 1.
- Only multiple assignments with date type 001, 010 or 020 can be substringed in a packed field, if a date type 002, 003, 004, 005, 006, 007, 011, 012, 015, or 017 is present in the field; date types 001, 010, or 020 cannot be assigned.
- Only multiple assignments with date type 002, 003, 004, 005, 006, 007, 011, 012, 015, or 017 can be substringed in a packed field, if a date type 001, 010, or 020 is present in the field; other date types cannot be assigned.
- You cannot assign a packed field with a field length of 5 as date type 005 and expansion type 2.

- You cannot assign a packed field with date position 1 and expansion type 2.
- Expansion type 2 is only valid for packed fields.
- A year length of 3 is valid only for date type 001 (year); in this case, only the expansion types 0, 1, and 4 are allowed.
- For date types 010 and 020, the expansion type must always be set to 1; BYPASS2000 automatically selects this value and displays an informational message.
- You can only assign one date with expansion type 2 in a packed field.
- You can assign binary fields only as standard date types. Substrings are not allowed. If the field is longer than the standard date, the latter is aligned to the right.
- You cannot assign binary 3-digit years.

Specifying Date-Field Expansion and Propagation

For each date field that you assign, you must specify how BYPASS2000 should expand and propagate it. The following options are available:

Default expansion (Option '0')
: Assigns the field as a date field and specifies that it should be expanded or not, according to the default conversion style that you have chosen for your application.

Not expand (Option '1')
: Assigns the field as a date field and specifies that it should not be expanded.

Expand + one reusing (Option '2')
: Assigns the field as a date field and specifies that it should be expanded by one digit.

  This setting applies only to packed fields where the added century flag can occupy the first unused half byte.

Expand + one (Option '3')
: Assigns the field as a date field and specifies that it should be expanded by one digit.

Expand + two (Option '4')
: Assigns the field as a date field and specifies that it should be expanded by two digits (even if the field already contains a century flag).

Propagate (Option '0')
: Assigns the field as a date field and specifies that it should be propagated.

  BYPASS2000 continues searching the propagation tree below this date field for related fields and identifies them as date fields. If conflicting information is received, these fields are left in a generic state that you must address manually.

Not propagate (Option '1')
: Assigns the field as a date field and specifies that it should *not* be propagated. This setting is not valid for database fields.

  Below this date field, BYPASS2000 stops searching the branch of the propagation tree for related fields. In this case, BYPASS2000 will identify related date fields only if it also finds them on another branch of the propagation tree.

Unidentified date fields will cause an incomplete conversion of your application. It is therefore important to check the propagation trace to make sure that no date fields have been overlooked.

Propagate upon confirmation (Option '2')
Assigns the field as a date field and specifies that it should only be propagated upon confirmation. This setting is not valid for database fields.

BYPASS2000 continues searching the propagation tree below this date field and for every related field prompts you to assign the field as a date field or not.

Note: If you decide to change the conversion style for your application, BYPASS2000 will treat the assigned fields according to the expansion and propagation settings that you have specified.

Restrictions on Year Length

If a 3-digit year (with century flag) in a field is expanded to become a 4-digit year, it will not lose its century flag. Instead, the year portion will be expanded to 5 digits (097 becomes 01997 instead of 01997). BYPASS2000 then ignores the century flag and considers the date as containing a 4-digit year.

Expanding Fields with Decimal Positions

When a field with decimals positions contains dates, the ADD/REMOVE routines act only on the integer part. For example, both 281198.000 and 281198.034 become 28111998.000.

Importing or Exporting Date Information

The following list shows the various import and export options on the BYPASS2000 Date-Field Assignment menu.

11. Import external field assignment (HSDATDFI)
Use this option to import external field assignment from the HSDATDFI file into the active conversion environment. This assignment is compatible with a BYPASS2000 V3R1M1 conversion environment.

See "Importing Date-Field Assignment from HSDATDFI" on page 117 for a detailed description of this option.

12. Import external field assignment (HSDATDFN)
Use this option to import external field assignment from the HSDATDFN file into the active conversion environment. This assignment is compatible with a BYPASS2000 V3R1M2 conversion environment.

See "Importing Date Field Assignment from HSDATDFN" on page 118 for a detailed description of this option.

13. Load field assignment into HSDATDFN for export
Use this option to export the current field assignment into the HSDATDFN file and reuse the date-related information in another conversion environment.

See "Creating HSDATDFN from an Existing BYPASS2000 Environment" on page 119 for a detailed description of this option.

14. Create interface from IBM SEARCH2000 repository
Use this option to create the HSDATDFN file by using information about date fields in database files that was collected by IBM SEARCH2000.

See “Creating HSDATDFN from the SEARCH2000 Repository” on page 120 for a detailed description of this option.

Working with Dates

The following list shows the various options on the BYPASS2000 Date-Field Assignment menu that you can use to work with dates.

6. Print list of assigned date fields
Use this information to verify the work that you have done, and to make sure that BYPASS2000 has all the year-sensitive information for each physical file. You can print the list of the fields that have been assigned.

7. Work with user-default date
You can instruct BYPASS2000 to always consider a field to contain a date (or a non-date) if it has a specific length and data type. This is equivalent to assigning (or locking) the specified field. BYPASS2000 will set the specified date type in every program, unless it finds a different assignment for the field in the program itself.

8. Work with date fields not to expand or propagate
You can instruct BYPASS2000 to treat certain fields as display or printer areas. You must provide this information when BYPASS2000 cannot correctly identify work areas that are used as display or printer areas.

10. Work with dates in files
You can display the list of dates contained in each physical file, and obtain information about the absolute displacement of each date in each record layout.

21. Find fields that are shared between files and device files
Use this option to find fields that are defined both in files and in device files (DSPF, PRTF).

Some programming techniques cause the expansion of a device-file field when a file field is expanded. Modify the sources that are involved, to prevent the expansion of the device-file field.

Assigning Date Fields in Display and Printer Files

Users who are currently keying 2-digit year information into date fields may not need to switch to a 4-digit format. Making changes to allow two additional digits for screens and reports is not only time-consuming, it is also impossible to automate completely.

Leave screen and report layouts unchanged, and add program logic to bridge between 2-digit and 4-digit years. BYPASS2000 automatically generates windowing logic based on user-specified date windows.

See “Choosing the Right Conversion Parameters” on page 8 for a discussion of the conversion parameter `Minimum YEAR value for 20th century` that lets you tailor windowing logic to the needs of your application.

Assigning Date Fields in Local Data Areas (LDA)

BYPASS2000 is able to assign date fields in data structures but it does not handle LDAs. You can assign data structures that are used to read from and write to LDAs, but you must modify the LDA length manually.

Redefining or Deleting Assigned Fields

You can modify the status of assigned fields after the initial assignment. The definition of the assigned fields is contained in one or more of the following files located in the *xxxx*DB library:

- HSDATFLD
- HSDATSDT
- HSDATLCK
- ANDATFIL
- HSDATFIL
- USDATNOE
- HSDATTYD.

When you delete a file from the conversion environment (after the file has been analyzed), BYPASS2000 automatically deletes the date-field assignment related to this file.

When you delete the definition of an assigned field in a source program and then run the analysis again, BYPASS2000 automatically deletes the assignment that was propagated from this field. See “Deleting the Assignment of a File Field” on page 115 for more information.

Date Propagation

Once all date fields in the database files have been assigned, BYPASS2000 can locate other date-related fields throughout your application. This step is called date propagation.

The propagation engine makes use of the network of relationships between storage areas that was built earlier, during the memory-level analysis. BYPASS2000 starts from the information that you have provided during the date-field assignment and propagates the year-sensitive attribute of the assigned date-fields through the internal program fields.

When only two programs are related, BYPASS2000 analyzes passed parameters and automatically extends the propagation to called programs. When a program calls more than one program using the same parameter call, you must handle this situation manually. (See “Propagating Date Fields When a Program Calls Multiple Programs” on page 26.)

See “Chapter 13. Propagating Date Fields” on page 123 for the steps that are required to propagate date fields throughout your entire application.

Considerations for Date-Field Propagation

In the BYPASS2000 data model of an application, each program is made up of a network of nodes and links which is built during memory-level analysis. The data model includes relationships between programs, as well as within programs. This is important, because some dates may only enter a program by a passed parameter.

BYPASS2000 uses the information of whether a field is date-related or not as input data for the propagation analysis. During the propagation analysis, BYPASS2000 follows the network of nodes and links, and propagates the date attribute from areas that correspond to fields that you have identified as being year-sensitive to all related areas in memory. It does so by setting propagation flags.

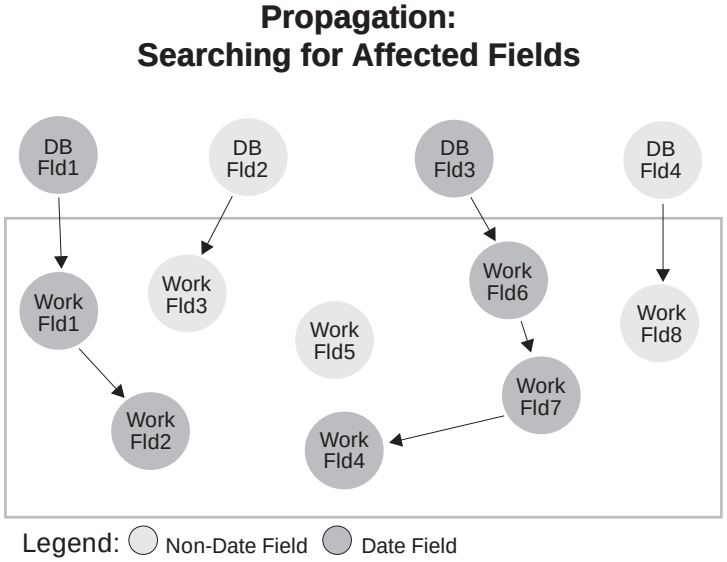


Figure 4. The propagation analysis phase

Ideally, all database storage areas should be flagged as belonging to one of two categories: year-sensitive areas or non-year-sensitive areas. However, BYPASS2000 will most likely encounter situations during propagation where a physical file field shows an unexpected date format in comparison to the corresponding database field. In this case, the following exists:

- An *incongruent* relationship exists between the field in the physical file and the corresponding database field.
- A *generic* relationship exists between the field in the physical file, and other fields that are related to it further down the propagation tree.

The application expert must address all incongruences found by BYPASS2000, before you can repeat the propagation to resolve the generic relationships.

Handling Generic Fields

When there is an incongruent relationship between fields in physical files and database fields, BYPASS2000 cannot decide whether related fields further down the propagation tree are year-sensitive fields or not. The status of these fields remains undecided or *generic*.

Generic fields may also be created inadvertently when you lock fields that have related fields further down the propagation tree. Locking a field prevents

BYPASS2000 to search for related fields in that branch. Any related fields that cannot be reached by any other branch of the propagation tree will remain generic. Therefore, these fields must be assigned manually.

You can prevent the creation of generic areas when locking fields by selecting **Lock with confirmation**. If you select this option, the propagation analysis explores the branch of the propagation tree below the locked field, and prompts you for every related field. You can then decide whether each of these fields should be assigned or not.

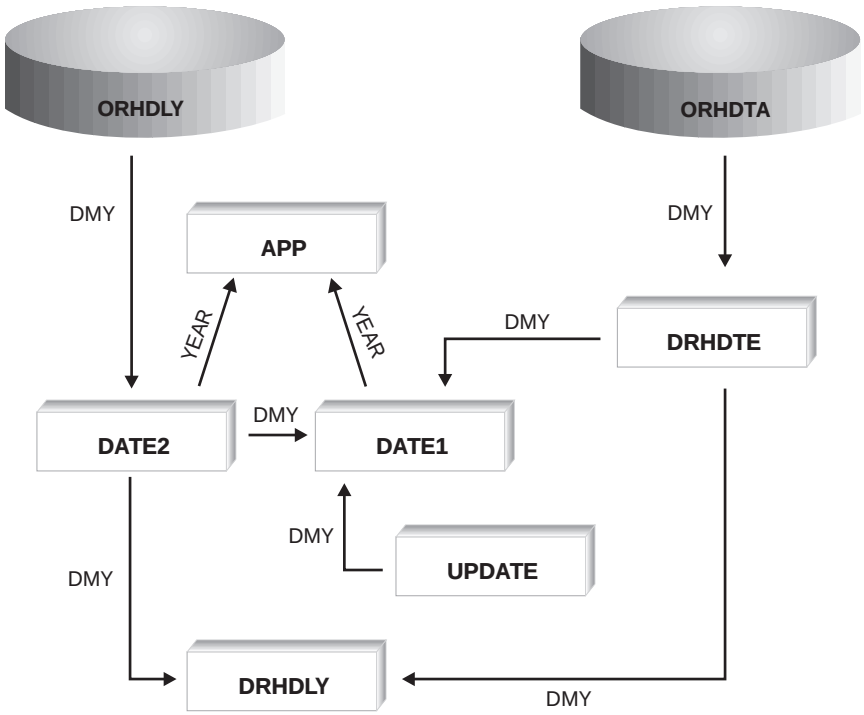


Figure 5. Date data which goes through the propagation analysis

Propagating Date Fields upon Confirmation

For each year-sensitive database field, you must decide whether this field should be propagated or not. If you decide not to propagate a particular year-sensitive field, the following will happen:

- When BYPASS2000 searches the propagation tree and encounters this field, the search will not continue further down this branch of the tree.
- Below this point, BYPASS2000 will not consider any fields for widening. Unless these fields can also be accessed from another branch of the propagation tree, BYPASS2000 will not acknowledge them as dates.

To allow the propagation analysis to reach all fields in the propagation tree, you can choose to propagate a particular field upon confirmation. In this case, the following will happen:

- When BYPASS2000 searches the propagation tree and encounters this field, the field itself will not be propagated, but the search for related fields will continue down this branch of the propagation tree.
- Below this point, BYPASS2000 inquires for each related field found whether you want to propagate it or not.

See “Chapter 13. Propagating Date Fields” on page 123 for a detailed description of the steps required to propagate date fields.

Propagating Date Fields When a Program Calls Multiple Programs

When a program calls more than one other program using the same parameter area, BYPASS2000 does not automatically propagate date fields. Having the same parameter call different programs may involve using a storage area that contains different date formats at different times. Because BYPASS2000 cannot determine in advance which format will be used in each call, it stops propagation at this point to avoid propagating incorrect date information.

If you check the conversion log of the caller program, you will see a message for each parameter that BYPASS2000 has not propagated. The solution is to manually assign the linkage parameters of called programs with the correct date formats.

You can also choose to force BYPASS2000 to propagate all date parameters in the caller program throughout all called programs. See “FREE-IPC” on page 62 for a description of the necessary steps.

Application Conversion

Once all areas that contain date-related information have been identified without missing or unclear information, BYPASS2000 can start converting DDS, CPY, SQL, and program source code to handle the new expanded dates.

During the conversion phase, BYPASS2000 highlights the following information:

- Changes made in the modified sources.
- Lines of code added.

  For example, code added for the removal of century information before output to a display file.
- Requests to manually verify and resolve logical incongruences.

  For example, an area that is used for both date and non-date data.
- Instructions that need to be checked for accuracy.

  For example, the comparison between a date and a constant equal to 00.

For more information, refer to “Chapter 16. Converting Source Code” on page 143.

Testing the Converted Application

Testing your converted application can be a challenge. It is difficult to simulate that the application is running in the future. Even if you were to change your system date for testing purposes (which you may not want to do, because it has many implications for application and system functions), testing with data that comes from current or past years may not adequately stress the application.

BYPASS2000 provides the following options to help make your testing efforts easier. For detailed instructions on generating test applications, see “Chapter 17. Testing the Converted Application Using Utility Programs” on page 147.

Generation of test data

BYPASS2000 can migrate your current files to their new expanded date formats. During this process, it can also add a specific number of years to the date values in each record. This provides you with a more realistic set of data for testing the application as if it was the later part of 1999 or even 2000.

Date shifting on system date

BYPASS2000 can intercept system date values that are retrieved by the application and modify their values to simulate running the application at a future point in time. See “Choosing the Right Conversion Parameters” on page 8 for instructions on how to enable date shifting.

Date integrity checking

BYPASS2000 can generate Date Integrity Modules (DIM) that check the value of the century portion of expanded dates in your files, each time a record is updated or created. This provides assurance that the newly expanded year fields are being updated with appropriate century values (either ’19’ or ’20’). DIMs are implemented as called programs via physical file triggers to ensure that they will be called on every update to the file.

Generating BYPASS2000 Utility Programs

Once you have converted your application, you can generate utility programs to help you migrate, test, and monitor your application’s databases.

See “Chapter 17. Testing the Converted Application Using Utility Programs” on page 147 for instructions on creating the following programs.

Migration programs

After the conversion of your application, you need to copy the records from your old files (2-digit year fields) to the new files (4-digit year fields). Migration programs insert routines to add the century into the migration code. BYPASS2000 creates one migration program for each converted data set.

Migration-dispatcher program

The migration dispatcher program ensures that the correct migration program is called. It saves you from having to write a separate CLP program to call each migration program.

Test-migration programs

The test migration programs move your application into the future, by shifting the dates in your database forward by a certain number of years. Use them to test how your application will behave at the turn of the century.

Test-migration dispatcher program

The test migration dispatcher program is similar to the migration dispatcher program. It calls the appropriate test migration programs to perform the year shift.

Date-Integrity Module (DIM) programs

The DIM programs verify the year-sensitive fields in a converted file before you update any records. These programs are called by the DIM dispatcher program.

Date-Integrity Module (DIM) dispatcher program

The DIM dispatcher program is called by a trigger program that you must link to the converted files. The trigger program passes the name of the converted file to the DIM dispatcher program, which starts the appropriate DIM program.

Chapter 3. Preparing Your Application for Processing

Before you can process your application with BYPASS2000, you must set up a conversion environment to provide BYPASS2000 with information about the location of all the sources and programs that make up your application. To allow BYPASS2000 to analyze the application with minimal manual intervention, ensure the following:

- All source files for the application are included in the conversion environment.
- All program names match the names of the corresponding runtime objects.
- No duplicate filenames exist for source members or runtime objects of the same type.
- All files specified on an OVRDBF command are available in the conversion environment.
- Unused parts are removed from your application.

Tip: Splitting a large application into smaller applications allows you to convert them separately.

While analyzing each of the smaller applications, BYPASS2000 requires access to *all* elements of the original application in order to properly build a complete image of the application. BYPASS2000 will expand only the date-related fields in the application being currently converted.

(See “Chapter 6. Setting up the Conversion Environment” on page 55 for detailed information.)

Considerations for System/36-Style Applications

Applications that contain files that are not externally described, such as S/36E-style programs, are good candidates for conversion with BYPASS2000. BYPASS2000 uses its own storage map to determine which fields are date fields. It can therefore make this determination reliably, even if a field name changes from program to program, or if the field disappears in the I-specs of some programs.

For this type of application, file fields typically have different names in different programs. In some programs, some of a file’s fields may not even appear. Not being able to rely on field names makes it difficult to manually change the application to correctly handle dates from different centuries.

Before BYPASS2000 can process an application where some or all of the files are not externally described, you must create external field descriptions. This will save time later on, because the external file-field descriptions will require less user input to determine the location of date fields in files.

Files which have more than one record format require special care. See “Files Having Multiple Record Formats” on page 33 for more information.

Once the preliminary steps are completed, BYPASS2000 can convert your System/36-style applications.

F Statement in COPY Redefining an F Statement in a Program

In System/36-style programs (or even older programs) it is possible to define an F statement in the program and subsequently include a call to a COPY that contains one or more lines of F statements. The meaning of this syntax is that the line in the COPY integrates the statement in the program. BYPASS2000 cannot understand the incomplete F statement of the COPY and therefore recognizes only the F statement in the program.

Considerations for RPG and COBOL Applications

The following sections provide information which you should consider when processing both RPG and COBOL applications.

Propagation of Arithmetic Statements

BYPASS2000 propagates the following arithmetic operations:

- A numeric value up to 9 can be added to a year.
- A numeric value up to 9 can be subtracted from a year.
- A date field can be multiplied by the numeric values 100, 1000, 10,000, 100,000, and 1,000,000.
- A date field can be divided by the numeric values 100, 1000, 10,000, 100,000, and 1,000,000.
- A date field can be multiplied by the numeric values 100.0001 and 10000.01 (used to reverse dates).

An arithmetic expression is considered to be the sum of its elementary operations: if a single term is either a year, or one of the operations listed above, the propagation works correctly. BYPASS2000 recognizes the division of a year by 4 as a leap-year calculation, and does not propagate it.

Some arithmetic operations on a date could change the position of the year in the field, depending on the value assumed by the operand used in the computation. In such a case, if BYPASS2000 cannot clearly determine the position of the year in the result, it generates a user inquiry during the propagation phase.

You must assign the operand if it contains year-related information.

Calls Using Program-Name Variables

BYPASS2000 uses the program names in call statements to create a link between the passed parameters and the linkage section or the entry parameter list of the called program. If the name of the called program is contained in a variable, BYPASS2000 will attempt to determine its proper value. If the name of the program cannot be retrieved, BYPASS2000 requests this information from the user.

If BYPASS2000 is able to determine the value of the called program, you may still need to confirm this information. (This is the case if you have specified `N` for the parameter ″Assume value of program-name variable was verified″ during the conversion environment setup, as described in “Choosing the Right Conversion Parameters” on page 8.)

When user intervention is required, the program analysis stops with the analysis flag set to 7, and BYPASS2000 generates a message requiring user information. Once you have provided the value that the variable used in CALL can assume, you must repeat the analysis step.

Calls to Multiple Programs Using the Same Parameter Area

When a program calls more than one program using the same parameter area, BYPASS2000 does not propagate date fields.

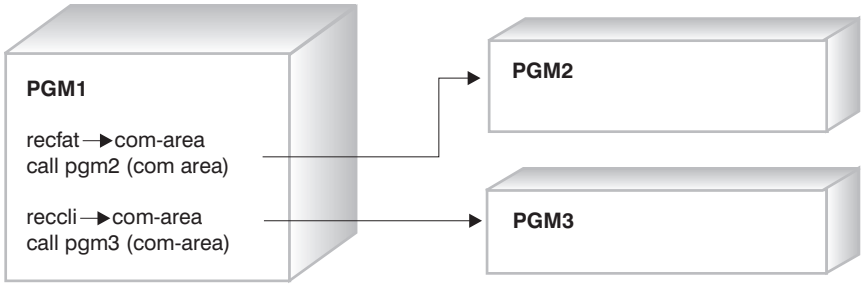


Figure 6. Not propagated fields

Without runtime analysis, BYPASS2000 cannot predict with certainty whether such a parameter area will always contain information in the same format, or whether the format will change during processing. To avoid the propagation of incorrect date information, BYPASS2000 stops propagation when it encounters this situation.

You must manually assign the linkage parameters of the called program, to ensure correct propagation results.

Number and Length of LINKAGE Parameters

BYPASS2000 assumes that there are the same number of LINKAGE parameters in a called program as there are parameters in the corresponding caller program, and that corresponding parameters have the same length.

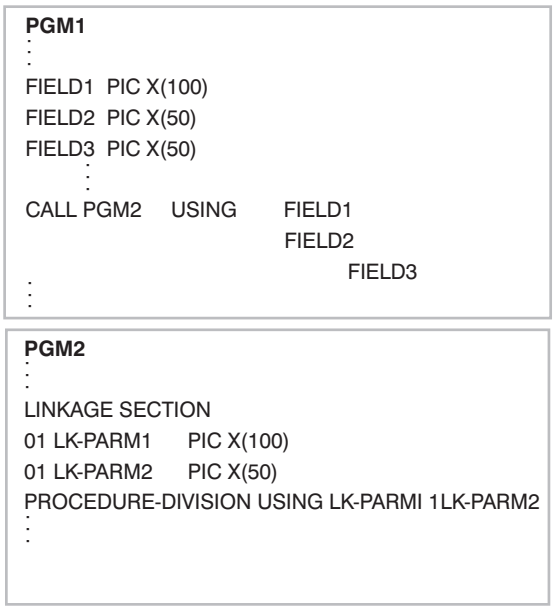


Figure 7. LINKAGE parameters displaying number and length

If your application is structured differently, you must manually assign the LINKAGE parameters. In the example illustrated above, you would manually assign the fields LK-PARM1 and LK-PARM2.

Data-Structure Definition Split Across Copybooks

Data structures must be defined in a single Copybook. BYPASS2000 cannot handle the case where the definition of a data structure is split, with one part defined in Copybook A and another part defined in Copybook B.

To circumvent this restriction, you should create a new Copybook that joins the two existing Copybooks into one source. This new Copybook must replace the two existing Copybooks in the program source. With this change, the data structure is defined in a single Copybook and will be processed correctly.

EXEC SQL in COPY Sources

BYPASS2000 can analyze EXEC SQL in a program source but not in a COPY source. If an EXEC SQL is found in a COPY, BYPASS2000 may produce unpredictable results.

BYPASS2000 is able to handle SQL statements in programs, even if the host variables are declared in a COPY. On the other hand, if the SQL statement is located in a COPY, BYPASS2000 may not be able to correctly determine all the relationships between the host variable and the SQL column.

Consider the following example:

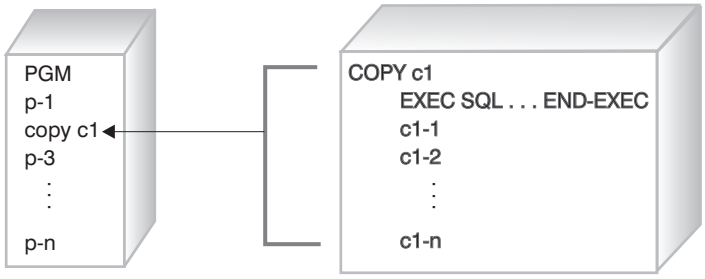


Figure 8. EXEC SQL in COPY source

You should expand the COPY source into the program source to ensure that all relationships between host variable and SQL column are found:

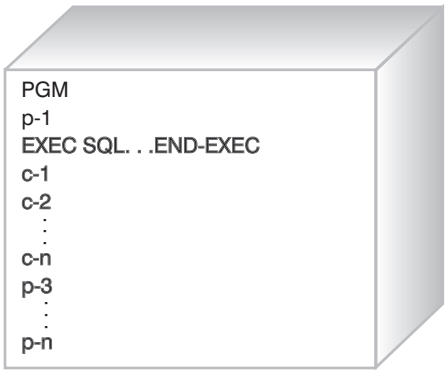


Figure 9. EXEC SQL in program source

LIKE in Field Definition

If the application refers to a field that is declared in a different source, BYPASS2000 does not handle the LIKE clause in the field definition. Therefore, you must replace the LIKE clause in the source with the real field definition.

SST and CONCAT Instructions in the DDS of Logical Files

BYPASS2000 does not automatically identify the SST and CONCAT keywords in the DDS of logical files. You must manually add the DDS of logical files to the conversion environment.

Files Having Multiple Record Formats

BYPASS2000 has special requirements for files that have multiple record formats. You must create a COPY for each record format (or a single COPY for all formats), containing the field descriptions, so that BYPASS2000 can generate the migration routines. You must assign these COPYs as well as the corresponding date fields in the program source to enable BYPASS2000 to correctly propagate the correct formats throughout the program.

Record-ID Position in Multiple-Format Files

When BYPASS2000 converts a COPY that is associated with a multiple-format file, or a program that uses an internal definition of a multiple-format file, it cannot correctly update the record ID position. You must update the record ID position manually.

Multiple-Format File with a Layout Described Using Indicators

If a multiple format file is defined using indicators like in the following example, BYPASS2000 is not able to recognize the file as a multiple format file. It issues a warning message and you must convert the source manually. You can also rewrite the statements in an equivalent format that BYPASS2000 supports.

```
FMULTIF    IP  F     128 13AI     4 DISK

IMULTIF  NS  01   2 C1

I       OR   02   2 C2

I       OR   03   2 C3

I                                      4   50FLDA

I                                      4   70FLDB L2

I                                      8  16 FLDC L1

I                                     17  56 FLDD1     01 **

I                                     17  81 FLDD2     02

I                                     17  25 FLDD3     03

I                                   P  26  280FLDE3     03
```

Missing Files

If a program uses a file that is not present in your conversion environment, BYPASS2000 issues an error message. You must add this file to your environment before BYPASS2000 can proceed with this program.

Missing Programs

If your application calls a program that is not present in your conversion environment, BYPASS2000 will not issue an error message. Make sure that all programs are available in the conversion environment before processing your application.

Duplicate Source Names

BYPASS2000 requires unique names for all source to correctly identify the relationship between data and source. Otherwise, BYPASS2000 loads the first source found, based on the list built during the conversion environment creation.

Note: BYPASS2000 builds an inventory of all source names and source types, *without* the qualifying library names. If you store multiple versions of a program in different libraries (for example, an older version, a customized

version, and a saved version), the inventory list will show multiple identical source names. In this case, it is impossible for BYPASS2000 to relate data to the correct source.

Consider the following example. You must decide which source BYPASS2000 should process, and remove all other sources having the same name.

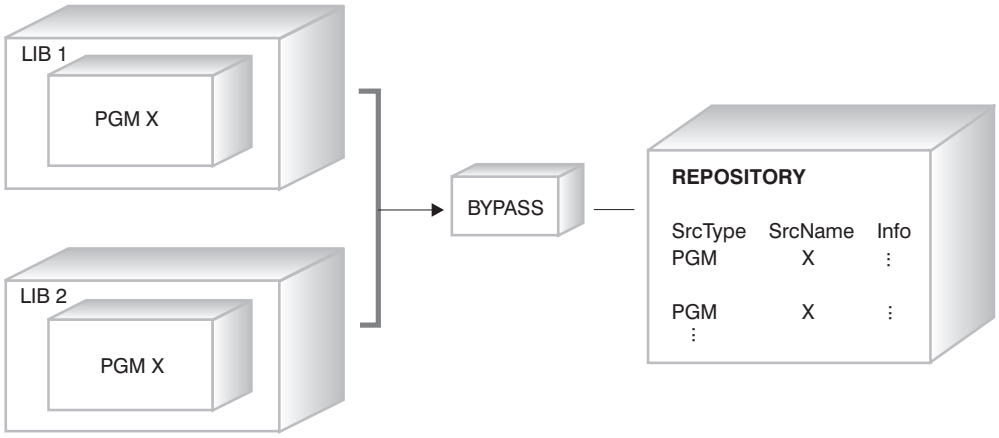


Figure 10. Multiple sources sharing the same name

Program Source Names Do Not Match Program Object Names

BYPASS2000 uses the program names in call statements to understand and map inter-program relationships between calling programs and called programs. If the names of program objects do not match those of the corresponding program source, BYPASS2000 will not be able to find the correct parameter definition.

Consider the following example:

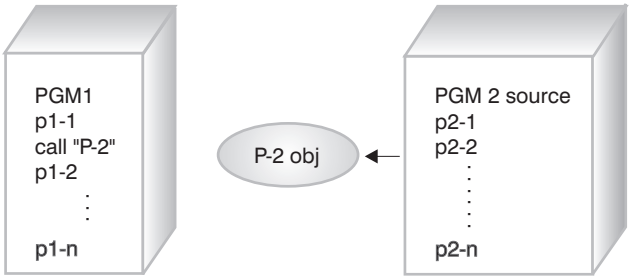


Figure 11. Source and object names which do not match

You must rename the program source to match the name of the program object:

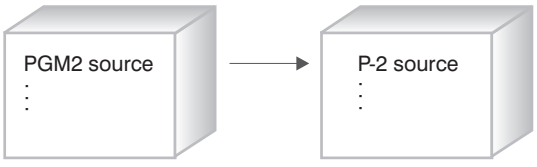


Figure 12. Program source which is renamed

Table Size Limit

When very large tables or arrays are used in COBOL programs, there may be a need for some manual modification in the converted source. To limit space requirements and improve performance, BYPASS2000 stores only some of the occurrences when large arrays are defined.

You should manually review the size of the target group area to ensure it is large enough to receive all the occurrences of data from the large table or array, specifically, if all of the following are true:

- The table or array contains an OCCURS value larger than 256.
- One or more dates are contained in the table or array data.
- The table or array is subsequently moved as a group to another area of storage in the program.

Variables in the CL CVTDAT Instruction

If your application contains a CL program that uses a variable instead of a constant as the parameter for the CVTDAT instruction, CVTDAT would only be fully resolved at runtime. To enable BYPASS2000 to correctly understand this instruction, you must replace the variable with a constant value before processing your application.

Year Values of 99 and 00

Year information that does not belong to a date and contains the values 99 or 00 is always converted to 1999 or 2000. BYPASS2000 does not consider them to be special values, and does not convert them to 9999/0000.

Migration of Packed Fields

BYPASS2000 cannot not migrate packed fields that contain more than one year substring.

Environment Creation

Files are created with the initial and incremental expansion restrictions that are equal to the default values of the AS/400 on which BYPASS is running. If such restrictions are not sufficient to include the data that is necessary for the conversion process, you must manually expand the files that may possibly be affected.

Inter-Program Propagation

The parameters of both the caller and the called programs must be the same in terms of number, length and type. This applies to all languages. If this is not the case, you must assign these parameters manually

Considerations for RPG Applications Only

The following considerations apply to RPG only.

Fields Used to Reverse a Date

If a field is used as an area to reverse a date (for example, from YMD to MDY), you must assign this field in two different formats, and assign it as not-to-be expanded and propagated. This may result in generic fields, which require manual assignment.

Note: If the name of this field is shared by a database field, this double assignment is not possible. The database field must be assigned with a single format, and the related field in the program must be assigned manually in both formats.

Fields in Printer or Display Files Imported with COPY DDS

If the fields in the display or printer files are not declared in the program, but are imported with a COPY DDS, you must rename the imported fields. In this case, you may encounter problems when making changes to the source code. If the printer or display file is written during the RPG cycle, the write instruction is implicit, and the correct placement of the MOVE statement can be difficult.

- If the output file is a display file, place the MOVE statement where the field is used.
- If the output file is a printer file, insert the MOVE statement in the level break.

RPG Statements that Require Manual Handling

The following RPG statements are not automatically handled by BYPASS2000:

- For CAT statements, you must confirm propagation.
- SUBST position and length are not automatically adjusted . You must modify the source after the conversion.
- For PARM implicit moves, WHxx, DOUxx, and DOWxx, BYPASS2000 cannot adjust the relationship between short and long year-sensitive fields. It notifies you of the problem, and you must change the source after conversion.
- A MOVE between complex areas that have non-matching expansion types is currently not supported. You should change the source after conversion.
- KLIST with improper field length is not automatically adjusted. BYPASS2000 notifies you of the problem, and you must change the source after conversion.

Fields with the Same Name in Databases and Printer and Display Files

BYPASS2000 cannot automatically handle a date-data item that is used by the same name both as a database field (to be expanded), and as a field in a display or printer file (not to expanded). If such a situation occurs, you may decide to manually solve the problem in one of two ways:

1. Change the original source code before the conversion.
   a. Declare new fields for shared dates in display and printer files.
   b. Insert MOVE statements from, or to, the display or printer files.
      - Insert MOVE statements from the new fields to the original fields after READ statements involving shared date fields.
      - Insert MOVE statements from the original fields to the new fields before WRITE statements involving shared date fields.

With this approach, BYPASS2000 expands the original fields, and does not expand the new fields.

2. Change the converted environment after the conversion.
   a. Rename fields that share the same name in display and printer files.
   b. Change the converted code by manually inserting MOVE statements and code to add or remove the century.

If the fields in the display or printer files are not declared in the program, you must rename the imported fields. In this case, you may encounter problems when making changes to the source code. If the printer or display file is written during the RPG cycle, the write instruction is implicit, and the correct placement of the MOVE statement can be difficult:

- If the output file is a display file, you must place the MOVE statement where the field is used.
- If the output file is a printer file, you should insert the MOVE statement in the level break.

Standard Record Length in the Source

BYPASS2000 expects your application to use a standard record length. By default, BYPASS2000 generates source files with a record length of 92. RPGLE source files are created with a record length of 112.

BYPASS2000 is able to handle source files with a maximum length of 192 characters (any additional characters are lost).

Note: CLP sources are handled only if the source file has a length of 92 characters.

If your application uses a different record length, you must change the source files that contain the converted code in the new source library after the conversion of your application.

You must create new source files in the xxxxY2K library (or your equivalent new source library). The names of the new files must match those of the original files in the new source library (i.e. QRPGSRC, QDDSSRC, QCLPSRC). Failing to do so may cause unexpected results. Use the CHGSRCPF (Change Source Physical File) command to make the change.

Fields defined as *LIKE DEFN of Fields in COPY

During the memory-level analysis, BYPASS2000 correctly handles fields that are defined in the following ways as *LIKE DEFN of fields defined in a COPY:

- Field defined in the I (DS) specifications
- Field defined in the C specifications
- Field defined as *LIKE DEFN of a field that is defined in the same COPY

It does not handle fields that are defined as *LIKE DEFN of a field that is contained in a different COPY.

I Statements Related to an Externally Defined File

If a program contains I statements that refer to an externally defined file (for instance to rename fields in a way similar to KRENAME in F statements), BYPASS2000 treats the I statements as if they were related to an internally defined file. It therefore loses possible redefinitions of the fields and assigns ANDATRPF incorrectly.

LOKUP Instruction

BYPASS2000 does not convert an instruction that follows the pattern FIELD LOKUP ARRAY,IDX, where FIELD is a date to be expanded and ARRAY is assigned as a date not to be expanded. It issues a message with a severity level of 50 to inform you that you must convert the source manually adjusted (for instance, you could assign a place-holder field as not to be expanded).

The problem lies in the fact that place-holder fields on IF instructions are generated exclusively by applying an ADD instruction to the shortest field, and not by applying a REMOVE instruction to the longest one.

MOVEA Instruction

BYPASS2000 cannot completely handle the conversion of the MOVEA statement between date fields. The following statement

```
C* MOVEA AA,1 BB,1
```

where AA and BB are defined as follows:

- E AA 4 8
- E BB 4 6

and where

- AA is seeded as DMY with expansion type 4
- BB is seeded as DMY with expansion type 1

will be converted as follows:

```
B2OLDC* MOVEA AA,1 BB,1
B2CHKC MOVE '008X3' HB2§FA
B2CHKC MOVE AA,1 HB2§F
B2CHKC MOVE '006X3' HB2§TA
B2CHKC EXSR HB§RMV REMOVE CENTURY
B2CHKC MOVE HB2§T BB,1
B2INF *CVR2001 sev.20 Converted code may contain conceptual errors. Please
B2INF * verify.
```

You must manually change the converted source, for example by coding a DO cycle as follows:

```
B2OLDC* MOVEA AA,1 BB,1
C DO 4 I 10
B2CHKC MOVE '008X3' HB2§FA
B2CHKC MOVE AA,I HB2§F
B2CHKC MOVE '006X3' HB2§TA
B2CHKC EXSR HB§RMV REMOVE CENTURY
B2CHKC MOVE HB2§T BB,I
C END
```

Partial Sub-Definitions

In the case of a sub-defined structure where the sub-definition contains a century field but not a year field, BYPASS2000l does not understand that the year is already Y2K compliant. The same applies if the century flag replaces the century.

In the following definition

```
I DS
I 1 140ITIME
I 11 120ICENT
```

ITIME is only partially redefined by ICENT. If, by propagation, ITIME contains a 4-digit year date in such a position that both century digits happen to be in ICENT, BYPASS2000 does not understand that the year (position 13 in ITIME) is preceded by a century, and performs a further expansion of ITIME.

To avoid this situation, you must assign ITIME as a 4-digit-year in position 11.

Considerations for COBOL Applications Only

The following considerations apply to COBOL only.

COBOL Sub-strings

BYPASS2000 correctly identifies and propagates substring instructions. During the conversion process, it marks these instructions as BP2SST if the following applies:

- The substring contains a date that needs to be expanded.
- The string portion that precedes the substring contains an expanded date.
- The substring parameters are variables and the field contains a date that is not to be expanded.

In these cases, you must manually change the code.

In all other cases BYPASS2000 marks the substring instructions as BP2000, and no intervention is required.

MOVE and IF Instructions for Fields that Contain Multiple Dates

If BYPASS2000 encounters a MOVE or an IF instruction involving fields that contain multiple dates, it will not be able to generate the corresponding ADD-REMOVE instructions.

Consider, for instance, a MOVE instruction from a field that contains ten non-expanded dates to a field that contains the same ten dates, one of which one is expanded while the other nine are not. BYPASS2000 will not generate any ADD

instruction, but it will mark the instruction as BP2CHK. You must write the ADD instruction manually with the correct parameters.

In the case a MOVE CORRESPONDING instruction, instructions must ne generated for all elementary fields that contain a single date.

PERFORM Instruction

BYPASS2000 converts the PERFORM UNTIL VARIABLE > CONSTANT as if it were an IF instruction. If dates are involved, BYPASS2000 does not convert the PERFORM UNTIL VARIABLE > VARIABLE instruction, but marks it.

BYPASS2000 never considers the FROM clause within a PERFORM instruction. The instructions within the PERFORM instruction are converted as usual.

COMPUTE Statement

When you use a COMPUTE statement to set several fields within the same statement, BYPASS2000 generates message BPA0012 ″Unknown statement during analyis″. The following example illustrates this:

```
COMPUTE VAL-MP V352 (1) = - VAL-IMP
```

Different 01 Levels in FD Clause Sharing the Same Name

BYPASS2000 assumes that 01-level names are unique. It does not handle multiple 01 levels in the FILE section that share the same name qualified by different file-id names.

If BYPASS2000 encounters duplicate 01-level names, it issues a message to view the log and take manual action.

To address the situation, do the following:

1. Rename the duplicate 01 levels.
2. Delete the result of the previous analysis.
3. Run the analysis again.

Dynamic Reference Modification (Substring)

BYPASS2000 handles reference modifications only if the leftmost character position and the length are numeric constants. Therefore, you should manually assign the variables used in the statement.

More than Five Reference Modifications (Substring)

To improve performance, BYPASS2000 handles reference modifications only if they contain up to five constants and variables. You should change statements that use more than five constants and variables, and add the removed items again after the conversion has been completed.

Nested COPY

BYPASS2000 handles COPY statements in program source but not in COPY source (COPYs cannot be nested). For space considerations and performance reasons, BYPASS2000 allows only one level of nested PGMCOPY source.

Consider the following example:

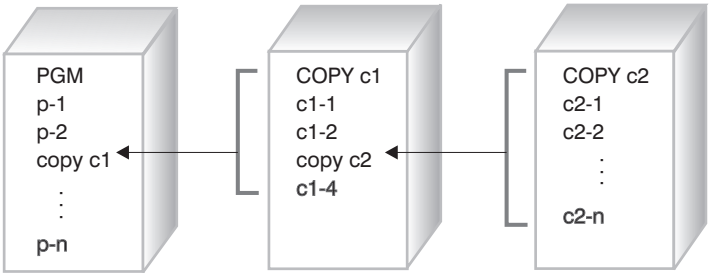


Figure 13. Nested COPY statements

You should replace all COPY statements found in COPY source with the corresponding COPY source:

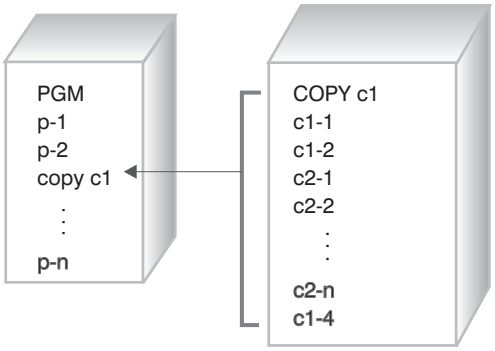


Figure 14. One level of a nested COPY statement

Conversion of COPY in PROCEDURE DIVISION

When a PROCEDURE DIVISION COPY is used by many different programs, BYPASS2000 will convert it as if it had been used only by one of them (selected randomly).

For instance, if a COPY contains the instruction MOVE A TO B, this COPY will be used by PGM1 (where A will be assigned as to be expanded and B will be assigned as not to be expanded), and by PGM2 (where A will be assigned as not to be expanded, while B will be assigned as to be expanded.). The converted COPY may contain either an ADD routine (as if it were used only by PGM2) or a RMV routine (as if it were used only by PGM1)

REDEFINES COPY

If the REDEFINES clause contains many non-expanded date fields, BYPASS2000 will mark only one of them as BP2000 (not necessarily the first one). It will ignore the remaining ones. If the fields are expanded, BYPASS2000 marks and modifies them.

Length of a Redefining Field

To reduce the complexity of some programs, BYPASS2000 assumes that a redefining field is not longer than the redefined field.

If your application contains REDEFINES clauses for which the redefining field is longer than the redefined field, you must make the following changes to your source code:

- The redefined field becomes the redefining field.
- The redefining field becomes the redefined field.

Position of REDEFINES Clause in Source

BYPASS2000 handles redefined fields at level 01 only if they are defined in the *previous* 01 level. During the analysis, a redefined field declared with a level greater than 01 is assumed to refer to the current 01 level.

If your application contains REDEFINES clauses for fields that are defined *before* their previous 01 level, you must make the following change to the source code:

1. Manually move the REDEFINES clauses to the position expected by BYPASS2000.
2. Delete the result of the previous analysis.
3. Run the analysis again.

Qualification-Level Limit (Nested OF)

For performance reasons, BYPASS2000 handles only a maximum of five qualification levels (five nested OFs). If your application uses more than five levels, you must change the qualifier list to only use the five levels that uniquely identify a variable, or change the variable names.

REDEFINES Clause with RENAME

When BYPASS2000 finds incongruences between redefined field formats, or if you force redefined fields to split, two or more I/O areas with the same name and different sequence numbers are generated.

- If the REDEFINES clause is at the 01 level, the field defined by the RENAMES clause is always associated with the *last* I/O area that was generated.
- If the REDEFINES clause is at a different level, the field defined by the RENAME clause is always associated with *all* I/O areas generated from the current 01 level.

If this association is not clear, you must handle the situation, as follows:

1. Replace the line that contains the RENAME by a new field definition with a REDEFINES clause to the same I/O area and the same characteristics as before. This can be done by, adding FILLER everywhere, except in the original renamed I/O area.
2. Delete the memory analysis of the program, as described in “Modifying Program Source” on page 140.
3. Repeat the analysis step, as described in “Running the Program Analysis” on page 85 .
4. Type `10` in the BYPASS2000 Memory-Level Analysis display to obtain additional user information, followed by option 3 (Work with Logical Redefines).
5. Select the program (option 1), and then the I/O area name (option 1).

   To assign the new field the same sequence number as the renamed I/O area, select the new field (option Y), and then select the renamed field (option 1).
6. Delete the memory analysis of the program again.
7. Repeat the analysis step.
8. Check the conversion log.

If BYPASS2000 does not split the I/O area, the REDEFINES clause is handled correctly.

REPLACING Clause in COPY DD Statements

BYPASS2000 recognizes the REPLACING clause in COPY DD statements for the following:

- Record format name
- Field name
- Generated level number
- Data type (except packed and binary).

If your application contains REPLACING clauses for other types, you must change the source.

REPLACING Clause in COPY (Working Storage and Procedure Division)

If a REPLACING clause is encountered in a COPY (this does not apply to COPY DD, DDS, DDSR, or DDR), BYPASS2000 will expand it internally, as the compiler would do. The following restrictions exist:

- BYPASS2000 expands a maximum of 100 replacements for COPY.
- BYPASS2000 expands a maximum of 10 words, for a maximum length of 30 for every single replacement. You must expand further replacements manually.
- If the resulting expanded line is longer than a COBOL source line, BYPASS2000 will not recognize it. It will leave the original COPY line unchanged and issue a message in the log. You must expand the COPY manually.

COPY REPLACING (Procedure Division)

BYPASS2000 correctly analyzes and propagates COPY REPLACING instructions that it finds in the Procedure Division. However, it does not convert them automatically. Therefore, you must convert them manually.

STRING Statements

BYPASS2000 does not automatically handle STRING statements. When it encounters STRING statements involving date fields and non-date fields, it issues a message to confirm the propagation effect. You must confirm whether or not to propagate these fields.

In the following example manually assign the fields DATE1, FIELD3 and YEAR, DMY.

```
1)
STRING  FIELD1     DELIMITED BY SPACE
        FIELD2     DELIMITED BY SIZE
        DATE1      DELIMITED BY SPACE
INTO    FIELD3

2)
STRING DAY
        MONTH
        YEAR       DELIMITED BY SIZE
INTO    DMY
```

Figure 15. STRING statements

Note: If year-sensitive fields used in a STRING statement are not automatically found during the propagation phase, you must assign them manually.

Subscript (Index)

For performance reasons, BYPASS2000 does not handle more than five subscription levels. You must change the index list to use only five levels. Add the removed indexes again after the conversion of your application is complete.

Year-Sensitive Fields with Initial Value

If your COBOL application uses year-sensitive fields with initial values, these fields will be expanded during the conversion. The century digits (e.g. 19 or 20) will not be added automatically. After the conversion, you will have to add the century digits manually to these fields.

Consider following example of the year-sensitive field WS-C-DATE before and after the conversion:

```
WS-C-DATE    PIC 9(6) VALUE 970516
```

is converted to:

```
WS-C-DATE    PIC 9(8) VALUE 970516
```

and must be manually changed to:

```
WS-C-DATE    PIC 9(8) VALUE 19970516
```

Chapter 4. Migrating an Existing V3R1M1 Environment Library

To migrate an existing BYPASS2000 V3R1M1 environment library to V3R1M2, perform the following steps:

1. On an AS/400 command line enter `ADDLIBLE QBP2000`.

   Tip: Skip this step if you have already added the BYPASS2000 product library to your library list.

2. Enter `BP2000 *GEN`.
3. Enter `BPMGRR2R3`. The Migration of V3R1M1 to V3R1M2 screen appears.

```
                  Migration of V3R1M1 to V3R1M2 (BPMGRR2R3)

Type choices, press Enter.

Environment library  . . . . . .   MYENVDB         Name
Run in batch . . . . . . . . . .   *YES          *YES, *NO

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

4. Specify the name of the environment library that you want to migrate, and press Enter.

Preserving the Conversion Style

If you want to achieve the same conversion result in V3R1M2 as in V3R1M1 for an existing conversion environment, specify conversion style 4 on the default conversion style parameter in the Customize Conversion Parameters screen shown below.

```
                    Customize Conversion Parameters


Conversion-database library  . . . . . . . . . . . . . . . . . . : MONAXXDB
System 36 COPY library . . . . . . . . . . . . . . . . . . . . . : *NONE
Default conversion style . . . . . . . . . . . . . . . . . . . . : 4
Column indentation: 1-9. . . . . . . . . . . . . . . . . . . . . : 4

Use caller-called relationships during propagation . . . . . . . : Y
Maximum level of propagation tree  . . . . . . . . . . . . . . . : 012
Maximum elapsed time for analysis of one program . . . . . . . . : 0003600
Maximum elapsed time for propagation of one program  . . . . . . : 0003600
Maximum elapsed time for conversion of one program . . . . . . . : 0003600
Remove REM marker from converted lines . . . . . . . . . . . . . : N
Assume that value of program-name variable was verified  . . . . : N
Insert marker on the right ('Y')instead of the left ('N'). . . . : N
Insert SHIFT instruction on system date  . . . . . . . . . . . . : Y
Generate instruction with quote('Y') instead of apostrophe('N'): Y



                                                                More...

F3=Exit        F5=Refresh        F12=Cancel        F9=Save
```

In BYPASS2000 V3R1M1, fields that you have assigned to be expanded have the expansion type 0. In V3R1M2, for conversion style 4, BYPASS2000 expands all fields assigned with expansion type 0, as well as all propagated fields. This behaviour guarantees the same conversion result in V3R1M2 as in V3R1M1.

Changing the Conversion Style

If you want to achieve a different conversion result for an existing V3R1M1 conversion environment, specify any conversion style other than 4 on the `Default conversion style` parameter in the Customize Conversion Parameters screen.

Based on the conversion style that you choose, BYPASS2000 interprets the existing expansion type of the assigned and propagated fields to achieve the desired conversion. Simply run the propagation phase again.

Preserving the Expansion Type

BYPASS2000 V3R1M2 introduces the following new date-field expansion types:

- Expand + one reusing
- Expand + one
- Expand + two

See “Specifying Date-Field Expansion and Propagation” on page 20 for a description of these expansion types.

To obtain the equivalent of the ″Expand″ setting in BYPASS2000 V3R1M1, specify the expansion type ″Expand + two″.

Migrating to BYPASS2000 V3R1M2 Halfway Through a Conversion

If you have used BYPASS2000 V3R1M1 to convert part of your application, you can migrate to BYPASS2000 V3R1M2 and convert the remainder of the application. BYPASS2000 V3R1M2 is fully compatible with BYPASS2000 V3R1M1.

Part 2. BYPASS2000 Tasks

Chapter 5. Starting BYPASS2000

To use BYPASS2000, the product library QBP2000 must appear in your library list. To start BYPASS2000 for the first time, do the following:

1. On an AS/400 command line enter `ADDLIBLE QBP2000`.

   Skip this step if you have already added the BYPASS2000 product library to your library list.

2. Enter `BP2000 *GEN`.

   The BYPASS2000 Environment Setup menu appears.

```
BP4AMBI                 BYPASS2000 Environment Setup
                                                          System:   TORAS002
Select one of the following:

     1. Create conversion environment

     3. Customize conversion parameters

     6. Work with object and source libraries
     7. Work with relationships between objects and sources
     8. Customize environment COPYs
     9. Work with field type
    10. Work with additional parameters
    11. Work with system fields

    13. Apply BYPASS2000 software key
    14. Create DDS from COPY
    15. Create COPY from file object
                                                                    More...
Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
(C) COPYRIGHT HAL S.p.A. 1994, 1998.
```

If you have not already done so, you must activate BYPASS2000 with a valid software key before continuing. See “The BYPASS2000 Software Key” on page xv for information on requesting and applying the software key.

If you want to migrate existing environments that were created with BYPASS2000 V3R1M1, see “Chapter 4. Migrating an Existing V3R1M1 Environment Library” on page 47 for instructions. Otherwise, proceed to “Chapter 6. Setting up the Conversion Environment” on page 55.

Working with an Existing Conversion Environment

If you want to start BYPASS2000 after having already created a conversion environment, do the following:

1. On an AS/400 command line enter `ADDLIBLE QBP2000`.

   Tip: Skip this step if you have already added the BYPASS2000 product library to your library list.

2. Enter `BP2000` *xxxxxx*`DB` where *xxxxxx* is the name of the environment. The BYPASS2000 for AS/400 Main Menu appears.

```
BP2000                BYPASS2000 for AS/400 - Main Menu
                                                          System:   TORAS002
Select one of the following:

     1. Environment setup
     2. Load database information
     3. Memory-level analysis
     4. Date field assignment
     5. Propagation analysis
     6. Application conversion

    70. Work with submitted jobs
    71. Work with all spooled files

    90. Sign off




                                                                       Bottom
Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
```

From here, you can access the high-level menus for all BYPASS2000 processing phases.

Chapter 6. Setting up the Conversion Environment

This chapter introduces and explains the BYPASS2000 menus and displays that are related to creating the conversion environment, and explains how to work with them.

The conversion environment provides BYPASS2000 with essential information about the location of libraries and sources, and with parameters that determine how BYPASS2000 will later convert date-related information throughout your application. Internally, the conversion environment is represented as an AS/400 library. This library is also called the BYPASS2000 conversion repository.

During the various BYPASS2000 processing phases, you may find that you need to add or modify files in the environment. See “Chapter 15. Changing the Conversion Environment” on page 139 for the required steps.

Creating the Conversion Environment

1. Type 1 in the BYPASS2000 for AS/400 Main Menu. The BYPASS2000 Environment Setup menu appears.

```
BP4AMBI                  BYPASS2000 Environment Setup
                                                          System:   TORAS002
Select one of the following:

     1. Create conversion environment

     3. Customize conversion parameters

     6. Work with object and source libraries
     7. Work with relationships between objects and sources
     8. Customize environment COPYs
     9. Work with field type
    10. Work with additional parameters
    11. Work with system fields

    13. Apply BYPASS2000 software key
    14. Create DDS from COPY
    15. Create COPY from file object
                                                                    More...
Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
(C) COPYRIGHT HAL S.p.A. 1994, 1998.
```

2. Type 1 to access the Create Conversion Environment display.

```
                  Create Conversion Environment (BPCRTLIB)

Type choices, press Enter.

Conversion identifier  . . . . .    _________    Name
Create default environment . . .    _________    *YES, *NO









                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

3. Specify the name of the conversion environment in the Conversion identifier entry field and specify whether or not you want to create the environment with default values. For more information see “Setting up a Conversion Environment” on page 13.
4. Press Enter to see the libraries that BYPASS2000 will use to set up the conversion environment.

```
                  Create Conversion Environment (BPCRTLIB0)

Type choices, press Enter.

Create default environment . . . > *NO           *YES, *NO
Original source library  . . . . > *NONE         Name, *NONE
Converted source library . . . . > *NONE         Name, *NONE
Converted object library . . . . > *NONE         Name, *NONE
Conversion-database library  . . > BPXXDB        Name
User-database library  . . . . . > *NONE         Name, *NONE
User-SQL-database library  . . . > *NONE         Name, *NONE
Run in batch . . . . . .   *YES          *YES, *NO









                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

5. Press Enter again and enter a job queue name.
   You can accept the default values, and press Enter.

Specifying Libraries for Objects and Source

Depending on whether you have created a default or a customized environment, the Work with Object and Source Libraries display shows different library settings.

If you chose the default setup, all library information is filled in the entries for the table that BYPASS2000 uses to access data and source files, as shown in the following display.

```
             Work with Object and Source Libraries

 Type choices, press Enter.                        Position to . . : ___________
   3=Copy    4=Delete     5=Display    *=Reset old informations

                                    Old Src Lib  Old Src File  Old Object Lib
Opt Type        Src SEU-Type   Status New Src Lib  New Src File  New Object Lib
 _  *FILE       DAT PF                 *NONE        *NONE         BP10OLD
                                       *NONE        *NONE         BP10NEW
 _  CBL         CPY CBL                BP10OLD      QCPYSRC       *NONE
                                       BP10NEW      QCPYSRC       *NONE
 _  CBL         CPY CBL38              BP10OLD      QS38SRC       *NONE
                                       BP10NEW      QS38SRC       *NONE
 _  CBL         PGM CBL                BP10OLD      QCBLSRC       *NONE
                                       BP10NEW      QCBLSRC       *NONE
 _  CBL         PGM CBL36              BP10OLD      QCBLSRC       *NONE
                                       BP10NEW      QCBLSRC       *NONE
 _  CBL         PGM CBL38              BP10OLD      QCBLSRC       *NONE
                                       BP10NEW      QCBLSRC       *NONE
                                                                        More...

 F3=Exit     F5=Refresh     F6=Create       Enter=Validate
 F12=Cancel  F17=Top        F18=Bottom      F20=One/Two Rows
```

If you chose a customized setup, the entries for Old Source Library, Old Object Library, and New Source Library are set to *NONE, and you must fill in the corresponding values.

```
             Work with Object and Source Libraries

 Type choices, press Enter.                        Position to . . : ___________
   3=Copy    4=Delete     5=Display    *=Reset old informations

                                    Old Src Lib  Old Src File  Old Object Lib
Opt Type        Src SEU-Type   Status New Src Lib  New Src File  New Object Lib
 _  *FILE       DAT PF                 *NONE        *NONE         *NONE
                                       *NONE        *NONE         *NONE
 _  CBL         CPY CBL                *NONE        QLBLSRC       *NONE
                                       *NONE        QLBLSRC       *NONE
 _  CBL         CPY CBL38              *NONE        QLBLSRC       *NONE
                                       *NONE        QLBLSRC       *NONE
 _  CBL         CPY UNS36              *NONE        QS36SRC       *NONE
                                       *NONE        QS36SRC       *NONE
 _  CBL         PGM CBL                *NONE        QCBLSRC       *NONE
                                       *NONE        QCBLSRC       *NONE
 _  CBL         PGM CBL36              *NONE        QS36SRC       *NONE
                                       *NONE        QS36SRC       *NONE
                                                                        More...

 F3=Exit     F5=Refresh     F6=Create       Enter=Validate      F9=Confirm
 F12=Cancel  F17=Top        F18=Bottom      F20=One/Two Rows
```

For each type of information that you will be converting, you must specify the location of the original (old) source files and the names of the original (old) object libraries, as well as the location for any converted (new) source files and objects:

***FILE** Type over the existing entries. Specify *xxxx*DAT as the Old objects library and *xxxx*NEW for the New objects library, where *xxxx* is a name of your choice.

If you decide not to use the default naming convention, you can use your own library names instead.

***CBL, CLP, DDS, OCL, RLE, RPG, and SQL**
Type over the existing entries. Specify *xxxx*OLD as the Old source library and *xxxx*Y2K for the New source library, where *xxxx* is a name of your choice.

If you decide not to use the default naming convention, you can use your own library names instead.

***ENV** Do not modify these entries. The *ENV type indicates an Environment COPY.

***SBR** Type over the existing entries. Specify *xxxx*Y2K for the New source library, where *xxxx* is a name of your choice.

The *SBR type indicates utility programs that BYPASS2000 can generate after the conversion of your application's sources.

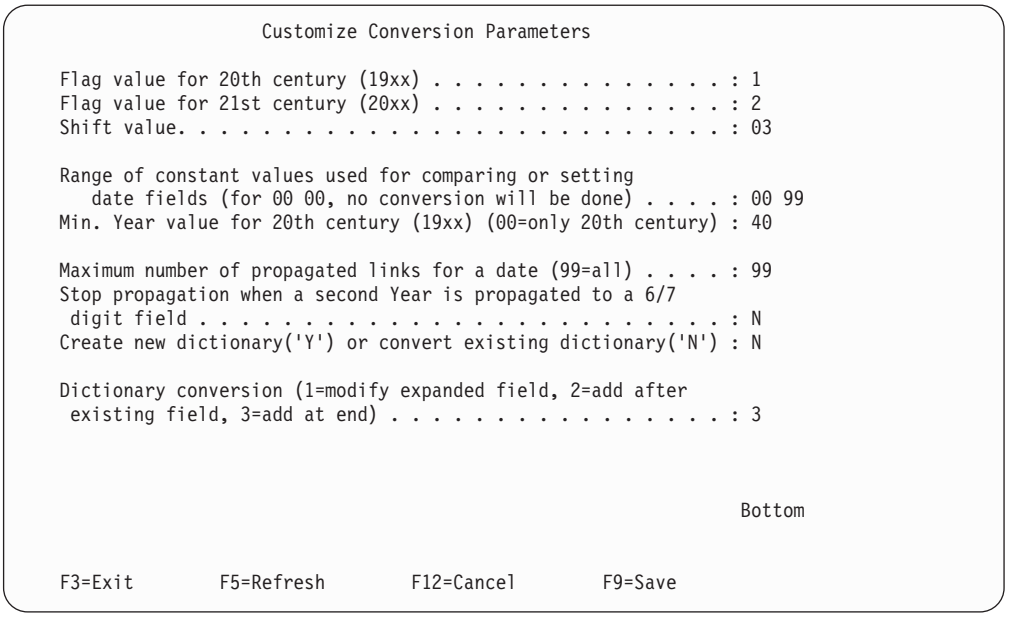
If you decide not to use the default naming convention, you can use your own library name instead.

Press F9 twice to confirm that you want to create the environment.

Customizing Conversion Parameters

Use the Customize Conversion Parameters display to specify the parameters that will control how BYPASS2000 converts your application.

```
                    Customize Conversion Parameters


Conversion-database library  . . . . . . . . . . . . . . . . . . : MONAXXDB
System 36 COPY library . . . . . . . . . . . . . . . . . . . . . : *NONE
Default conversion style . . . . . . . . . . . . . . . . . . . . : 4
Column indentation: 1-9. . . . . . . . . . . . . . . . . . . . . : 4

Use caller-called relationships during propagation . . . . . . . : Y
Maximum level of propagation tree  . . . . . . . . . . . . . . . : 012
Maximum elapsed time for analysis of one program . . . . . . . . : 0003600
Maximum elapsed time for propagation of one program  . . . . . . : 0003600
Maximum elapsed time for conversion of one program . . . . . . . : 0003600
Remove REM marker from converted lines . . . . . . . . . . . . . : N
Assume that value of program-name variable was verified  . . . . : N
Insert marker on the right ('Y')instead of the left ('N'). . . . : N
Insert SHIFT instruction on system date  . . . . . . . . . . . . : Y
Generate instruction with quote('Y') instead of apostrophe('N'): Y




                                                                More...

F3=Exit        F5=Refresh        F12=Cancel        F9=Save
```



You can access this display any time by selecting option 3 (Work with BYPASS2000 parameters) in the BYPASS2000 Environment Setup menu (option 1 of the BYPASS2000 for AS/400 Main Menu). See “Choosing the Right Conversion Parameters” on page 8 for a description of each conversion parameter.

The conversion parameters are stored in the BPPARM data area of the *xxxxxx*DB library. BPPARM also contains the list of the product and conversion libraries that are loaded when you issue the startup command BP2000 *xxxxxx*DB. See “Working with an Existing Conversion Environment” on page 53 for more details on the startup command.

See “Chapter 15. Changing the Conversion Environment” on page 139 for information on how to add components to the conversion environment or modify or delete existing components.

Displaying Field Types

To display the list of date field types that BYPASS2000 can handle in addition to the system defaults, type 9 from the BYPASS2000 Environment Setup menu.

```
                          Work with Field Type

 Type choices, press Enter.                        Position to . . :
   5=Display

                                        Short ----- Old ----- ----- New -----
Opt Cod Description                     Desc. T Len.  Int Dec T Len.  Int Dec  R
 _  001 YEAR                            YEAR  N 00002 002 000 N 00004 004 000  N
 _  002 YEAR,MONTH,DAY                  YMD   N 00006 006 000 N 00008 008 000  N
 _  003 DAY,MONTH,YEAR,/MONTH,DAY,YEAR DMY   N 00006 006 000 N 00008 008 000  N
 _  004 DAY-MONTH-YEAR/MONTH-DAY-YEAR   D-M-Y X 00008 006 000 X 00010 008 000  N
 _  005 YEAR AND JULIAN DAY             Y-JUL N 00005 005 000 N 00007 007 000  N
 _  006 MONTH,YEAR                      MY    N 00004 004 000 N 00006 006 000  N
 _  007 YEAR,MONTH                      YM    N 00004 004 000 N 00006 006 000  N
 _  010 CENTURY                         CENT. N 00002 002 000 N 00002 002 000  N
 _  011 COMPLEMENT YEAR                 C-YEA N 00002 002 000 N 00004 004 000  N
 _  012 COMPLEMENT YEAR,MONTH,DAY       C-YMD N 00006 006 000 N 00008 008 000  N
 _  015 COMPLEMENT YEAR,JULIAN DAY      C-JUL N 00005 005 000 N 00007 007 000  N
 _  017 COMPLEMENT YEAR,MONTH           C-YM  N 00004 004 000 N 00006 006 000  N
 _  020 CENTURY FLAG                    CF    N 00001 001 000 N 00001 001 000  +

 F3=Exit                    F5=Refresh      F12=Cancel
 F15=Load default records   F17=Top         F18=Bottom
```

The description and code for each standard date type are shown in the display. They are used during the date-field assignment phase described in “Chapter 11. Assigning Date Fields” on page 99.

You can use the PF15 key to reload the field types, if they have been accidentally erased from the table.

Changing the Parameter Table

You can change the values in the parameter table which BYPASS2000 uses for the application conversion. You can also create new parameter codes.

1. Type `10` on the BYPASS2000 Environment Setup menu to access the Work with Parameter Table display.

   This display lists the existing parameter codes.

```
                        Work with Parameter Table

                                                Position to . . :
 Type choices, press Enter.
   2=Change    4=Delete    5=Display

Opt Type     Code                              Seq.  Data
 _  BPCONST  99                                00000 9999                    ...
 _  BPCONST  9912                              00000 999912                  ...
 _  BPCONST  991231                            00000 99991231                ...
 _  BPCONST  99366                             00000 9999366                 ...
 _  BPCONST  9999                              00000 999999                  ...
 _  BPCONST  99999                             00000 9999999                 ...
 _  BPCONST  999999                            00000 99999999                ...
 _  BPENV    *ALL                              00000 All Environments        ...
 _  BPENV    OS400                             00000 OS/400                  ...
 _  BPLANG   *ALL                              00000 All Languages           ...
 _  BPLANG   CLP                               00000 OS/400 Control Language ...
 _  BPLANG   COBOL                             00000 COBOL                   ...
 _  BPLANG   RPG                               00000 RPG                     . +

 F3=Exit       F5=Refresh     F6=Create   F12=Cancel
 F15=Load defaults            F17=Top     F18=Bottom
```

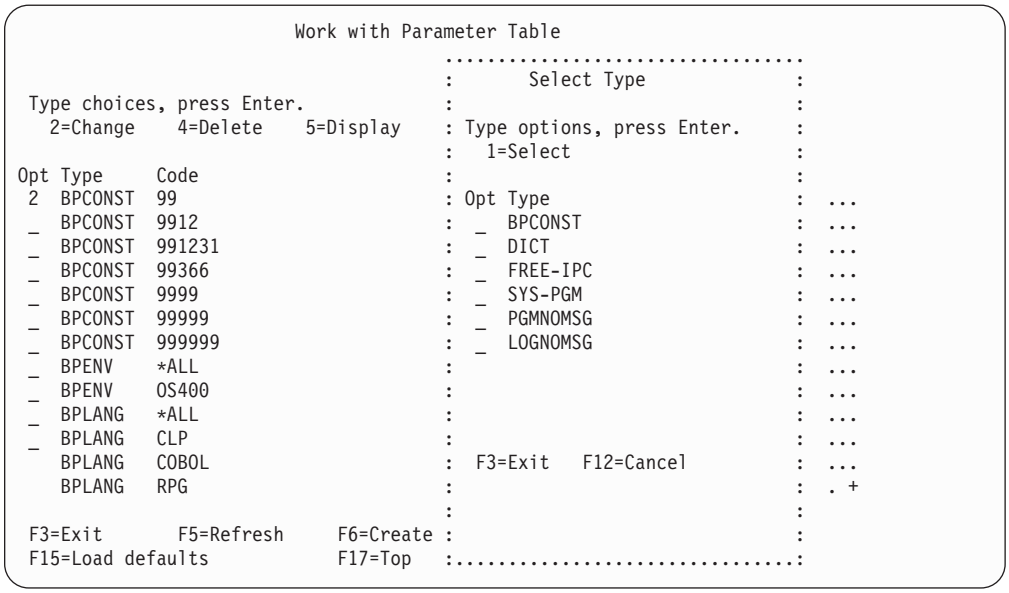
2. To change an existing parameter, type 2 next to an entry and press Enter.

   Note: You cannot change the parameter types BPENV and BPLANG.

```
                           Change Entry

 Type choices, press Enter.

 Type. . . . . . . . . . . . . : BPCONST

 Converting value. . . . . . . : 99

 Converted value . . . . . . . : 9999









 F3=Exit     F5=Refresh     F12=Cancel
```

   The converting value specifies the current value for a given constant that requires a particular conversion. If this value is used with a date, it will be converted based on your specifications. The converted value specifies the value of the special constant after the conversion.

3. Press F6 to create a new entry.



You can create one of the following parameter types:

BPCONST
Any special constant value.

DICT A file to be used as a dictionary.

FREE-IPC
Free inter-program propagation for a program.

When a program calls more than one program with the same CALL parameters, BYPASS2000 does not propagate these parameters throughout the called programs. A message in the conversion log will point this out. If you want to propagate the parameters, you can do one of the following:

- Assign the corresponding date field manually, in each called program.

  This approach may be time consuming, and can result in a large number of messages in the conversion log that must be acknowledged.
- Create a FREE-IPC parameter for the calling program.

  With this approach, all date parameters in the caller program will be propagated throughout all called programs.

SYS-PGM
System programs that should not be considered for inter-program propagation.

PGMNOMSG
Messages that BYPASS2000 should not insert into the converted code. By default, BYPASS2000 inserts messages.

(In order to create a new PGMNOMSG record, you must first remove the default record that shows *NONE in the Code column.)

LOGNOMSG
Messages that BYPASS2000 should not insert into the log. By default, BYPASS2000 inserts messages.

(In order to create a new LOGNOMSG record, you must first remove the default record that shows *NONE in the Code column.)

4. Type 1 next to an entry (FREE-IPC) in the Select Type window to access the Create New Entry display.

```
                          Create New Entry

Type choices, press Enter.

Type. . . . . . . . . . . . . : FREE-IPC

Program Name. . . . . . . . . : MYPROG

Additional description  . . . : Allows inter-program propagation of the CALL parameters









F3=Exit     F5=Refresh     F12=Cancel
```

5. Specify the program name and any additional information for this entry.

Adding System Fields

You can specify that BYPASS2000 should automatically insert a field with a specified length and date type as a date into your application. This field will be available in every program like other system date fields.

1. Type 11 on the BYPASS2000 Environment Setup menu to access the Work with System Fields display.

```
                          Work with System Fields

                                Position to date field:
Type choices, press Enter.
 2=Change    4=Delete     5=Display

Opt I/O-Area Name            Field Name              Environment Language
 _  *TIME                    *TIME                   *ALL        RPG
 _  *TIME14                  *TIME14                 *ALL        RPG
 _  DATE                     DATE                    *ALL        COBOL
 _  DATE                     DATE                    *ALL        CLP
 _  DAY                      DAY                     *ALL        COBOL
 _  PAGE1                    PAGE1                   *ALL        RPG
 _  QDATE                    QDATE                   *ALL        CLP
 _  SQLCOD                   SQLCOD                  *ALL        *ALL
 _  TIME                     TIME                    *ALL        COBOL




 F3=Exit           F5=Refresh    F6=Create          F12=Cancel   F17=Top
 F15=Load default system date    F19=Work with add. parameters   F18=Bottom
```

2. Press F6 to assign a date, with the same specifications and the same name, to a group of programs.

```
                          Create New Entry

Environment  . . . . *ALL       (TRK IN PARAM.TABLE = BPENV)
Language . . . . . . *ALL       (TRK IN PARAM.TABLE = BPLANG)
I/O-area name  . . .

Field attributes
Name . . . . . . . .  ____________________________
Displacement . . . .     0
Length . . . . . . .     0                          Int.    0  Dec.   0
Type . . . . . . . . X CHAR       ( /X/P/N/Z/B)     Sign    (Y/ )
Level  . . . . . . .  0                             Group   (Y/ )

Date attributes
Position . . . . . .     1
Length . . . . . . .     2
Date type  . . . . .   0
Year length. . . . . 2
Expansion style  . . 0   (0=Default 1=Not expand 2=Expand + 1 (packed
                         fields) 3=Expand +1 4=Expand +2)
Propagation type . . 0   (0=Propag. 1=Not propag. 2=Upon confirmation)
F3=Exit          F4=List of environments/languages    F5=Refresh
F12=Cancel       F19=Work with add. parameters
```

3. Specify the required parameters and press Enter.

   Note: You can assign a substring date field. However, multiple assignments for the same field are not permitted. Multiple assignment is only permitted for the date type ″YEAR″, with the year length in the date equal to 3. This will automatically be transformed into the Century Flag (CF) and Year date types.

4. Repeat the previous step for each general user-default date that you want to assign. You can also define non-date system fields (like TIME in COBOL). In this case the date type must be set to 0.

Creating an RPGHSPEC and DFTHSPEC Data Area

In order to simulate the presence of an RPGHSPEC or DFTHSPEC data area, do the following:

1. Press F6 in the Work with System Fields display.

```
                                                Create New Entry

Environment  . . . . *ALL       ALL ENVIRONMENTS
Language . . . . . . RPG        RPG
I/O-area name  . . . *RPGHSPEC

Field attributes
Name . . . . . . . . *RPGHSPEC
Displacement . . . .     1
Length . . . . . . .     4                              Int.    6  Dec.   0
Type . . . . . . . . P PACKED     ( /X/P/N/Z/B)       Sign    (Y/ )
Level  . . . . . . .  1                               Group   (Y/ )

Date attributes
Position . . . . . .     1
Length . . . . . . .     4
Date type  . . . . .   3 DAY,MONTH,YEAR,/MONTH,DAY,YEAR
Year length. . . . . 2
Expansion style  . . 1   (0=Default 1=Not expand 2=Expand + 1 (packed
                          fields) 3=Expand +1 4=Expand +2)
Propagation type . . 0   (0=Propag. 1=Not propag. 2=Upon confirmation)
F3=Exit          F4=List of environments/languages    F5=Refresh
F12=Cancel       F19=Work with add. parameters
```

2. Specify the desired date format in the Date type field. During the analysis, BYPASS2000 searches first whether there are H specifications in the RPG source. If no H specification is found, BYPASS2000searches for entry shown above. If no entry is found, BYPASS2000 assigns the standard date type.

   Note: You must specify *RPGHSPEC as the I/O-area name and as the field name, even if you create a DTFHSPEC data area.

Loading Source and File Objects

Note: If you have created a customized conversion environment, you can skip this section.

If you have set up your conversion environment with default values, you must now load the sources and file objects of your application into the appropriate libraries that BYPASS2000 has created for you:

If BYPASS2000 finds multiple files with the same name, you will receive a message that only the first file with this name has been loaded. You must resolve this situation before you can proceed to the next phase. All sources in the old source library must have unique names in order for the memory analysis to complete successfully. (See “Duplicate Source Names” on page 34.)

The sources of files and programs must be loaded into the old source library *xxxxxx*OLD. Load each type of source member into the appropriate file. An example is shown in the following table:

Table 5. Example of source files for source members

Source Type	Source File
DDS (PF, LF)	QDDSSRC
SQL (QMQRY)	QSQLSRC
CLP	QCLPSRC
RPG, RPGSQL	QRPGSRC
COBOL, CBLSQL	QCBLSRC

Table 5. Example of source files for source members (continued)

Source Type	Source File
COPY RPG	QCPYSRC
COPY COBOL	QLBLSRC
RPGLE, SQLRPGLE	QRPGLESRC

The following compiled file objects must be loaded into the old object library *xxxxxx*DAT:

- Database files (PF, LF)
- Device files (PRTF, DSPF)

Note: You can use empty copies of these files. You must eliminate duplicate file names before entering the next processing phase.

Changing Relationships Between Sources and Objects

If for a physical-file object BYPASS2000 cannot find a source with the same name in the inventory list you must handle this situation manually.

If the DDS source that defines the physical file has a different name, you can change the source/object relationships by doing the following:

1. Type 7 on the BYPASS2000 Environment Setup menu to access the Work with Relationships between Sources and Objects display.

```
               Work with Relationships between Sources and Objects
                                Select Parameters

Type choices, press Enter.

Library name:       *ALL          *ALL, name, generic*

Object name:        *ALL          *ALL, name, generic*









F3=Exit  F12=Cancel
```

2. Specify values for the library and object names you want to work with, and press Enter. A list of sources and objects appears.

```
              Work with Relationships between Sources and Objects

                                                    Position to . . :
 Type choices, press Enter.
   2=Change     4=Delete    5=Display
    -------- Objects --------  -------------- Sources -------------
Opt Typ Library    Name        Typ Library    File       Name        SEU Type
 _  DDS BP10OLD    ADSDEMO     DDS BP10OLD    QDDSSRC    ADSDEMO     PF
 _  DDS BP10OLD    ADU006P     DDS BP10OLD    QDDSSRC    ADU006P     PF
 _  DDS BP10OLD    ADU007P     DDS BP10OLD    QDDSSRC    ADU007P     PF
 _  DDS BP10OLD    ADU009P     DDS BP10OLD    QDDSSRC    ADU009P     PF
 _  DDS BP10OLD    ADU011P     DDS BP10OLD    QDDSSRC    ADU011P     PF
 _  DDS BP10OLD    ADU020P     DDS BP10OLD    QDDSSRC    ADU020P     PF
 _  DDS BP10OLD    ADU049P     DDS BP10OLD    QDDSSRC    ADU049P     PF
 _  DDS BP10OLD    CENDSP      DDS BP10OLD    QDDSSRC    CENDSP      PF
 _  DDS BP10OLD    CENDSP1     DDS BP10OLD    QDDSSRC    CENDSP1     PF
 _  DDS BP10OLD    MYPF        DDS *UNKNOWN   *UNKNOWN   MYPF        PF
 _  DDS BP10OLD    GETB01LA    DDS BP10OLD    QDDSSRC    GETB01LA    PF
 _  DDS BP10OLD    HHAVLX      DDS BP10OLD    QDDSSRC    HHAVLX      PF
 _  DDS BP10OLD    I00221      DDS BP10OLD    QDDSSRC    I00221      PF    More...

 F3=Exit     F5=Refresh     F6=Create     F10=Update relationships
 F12=Cancel  F17=Top        F18=Bottom
```

3. Type 2 for each object that shows *UNKNOWN as the source library and the source file specification. The Change Entry display appears.

```
                              Change Entry

 Objects
 Type  . . . . . . . DDS
 Library . . . . . . BP10OLD
 Name  . . . . . . . MYPF

 Sources
 Type  . . . . . . . DDS
 Library . . . . . . *UNKNOWN
 File  . . . . . . . *UNKNOWN
 Name  . . . . . . . DDS

 SEU
 Type  . . . . . . . PF
 Subtype . . . . . . PF

 User check  . . . . N
 Info. provider  . . USR


 F3=Exit     F5=Refresh     F12=Cancel
```

4. Specify which DDS source describes this object.

   Note: This step is only necessary for physical files that contain year-related information.

```
                              Change Entry

Objects
Type  . . . . . . . DDS
Library . . . . . . BP10OLD
Name  . . . . . . . MYPF

Sources
Type  . . . . . . . DDS
Library . . . . . . MYLIB
File  . . . . . . . MYFILE
Name  . . . . . . . DDS

SEU
Type  . . . . . . . PF
Subtype . . . . . . PF

User check  . . . . N
Info. provider  . . USR


F3=Exit     F5=Refresh     F12=Cancel
```

Customizing Environment COPYs

Use option 8 (Customize environment COPYs) on the BYPASS2000 Environment Setup menu to create copies of the BYPASS2000 environment COPYs.

```
                    Customize Environment COPYs (BPCUSTENV)

Type choices, press Enter.

Environment library  . . . . . .   *LIBL         Name, *LIBL, *CURLIB
Flag value for 20th century  . . > 3            Number
Flag value for 21st century  . . > 4            Number
Shift value  . . . . . . . . . . > 0            Number
Min. Year value for 20th centu   > 50           Number
Replace member . . . . . . . . .   *YES         *YES, *NO
Run in batch . . . . . . . . . .   *YES         *YES, *NO









                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

This option automatically copies the environment Copybooks from the QBP2000 product library to your development library. It performs the following tasks:

1. Analyzes all COPY members contained in the objects QLBLSRC, QRPGSRC, QRPGLESRC, QS36SRC, QS38SRC, and QCBLSRC (type *FILE) in library QBP2000.
2. Analyzes the libraries that are specified in the fields NEWSRCLIB and NEWSRCFIL of records with source CPY and type CBL, RPG, RLE, S36. You can modify these libraries in the Work with Object and Source Libraries display.
3. Copies the members of QBP2000 to the new libraries, replacing the default values for shift, century flag, and minimum YEAR for 20th century.

You can customize the environment Copybooks by setting the following parameters:

- Values for the century flag

  These parameters specify the value of the century flag that identifies the years between 1900 and 1999 (20th century) and between 2000 and 2099 (21st century).
- Shift value for system date and database test migration.

  This parameter specifies the number of years that should be added to system dates when there is a shift instruction. The migration programs will add this same number of years to database dates.
- Minimum YEAR value for the 20th century

  This parameter specifies the boundary year for ″windowing″ logic. Below and up to this this value, the Year is attributed to the 21st century. Above this value, the Year is attributed to the 20th century.

Note: To customize environment COPYs, you must specify your development library as type *ENV. To do so, type 6 on the BYPASS2000 Environment Setup menu to access the Work with Object and Source Libraries display.

Creating DDS from COPY

If you have an existing COBOL, RPG, or ILE RPG COPY source that defines a file layout, you can generate the corresponding DDS source automatically.

1. Type 14 on the BYPASS2000 Environment Setup menu to access the Create DDS from COPY display.
2. Fill in the required parameters and press Enter.

```
                      Create DDS from COPY (UTCRTDDSCL)

Type choices, press Enter.

To DDS . . . . . .                    Name
To file  . . . . . .                  Name
To lib . . . . . .                    Name
From COPY  . . . . . .                Name
Run in batch . . . . . .   *YES       *YES, *NO

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Creating COPY from File Object

You can create a COBOL, RPG, or ILE RPG COPY from a database file, starting from an existing DDS.

1. Analyze the database file. (See “Analyzing Databases” on page 79 for a description of the required steps.)

2. Type 15 on the BYPASS2000 Environment Setup menu to access the Create COPY from File Object display.
3. Fill in the required parameters and press Enter to create the COPY.

```
                  Create COPY from File Object (UTCRTCPYCL)

Type choices, press Enter.

To COPY  . . . . . .                    Name
To file  . . . . . .                    Name
To lib . . . . . .                      Name
Language . . . . . .                    Character value
From file  . . . . . .                  Name
From lib . . . . . .                    Name
Run in batch . . . . . .   *YES         *YES, *NO

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Creating User Options

You can create customized user options that you can use from various ″Work with....″ displays.

1. Access one of the following displays:
   - Work with Database Information
   - Work with COPY Sources
   - Work with SQL Sources
   - Work with Program Sources
2. Press F16 to access the Work with User Options display.

```
                              Work with User Options
                                                   Position to . . :
Type choices, press Enter.
  2=Change   3=Copy    4=Delete    5=Display


Opt. Code   Description                                        Command
```

```
 F3=Exit   F5=Refresh   F6=Create   F12=Cancel   F17=Top  F18=Bottom

No data to list
```

3. Press F6 to create a new user option.

```
 Insert New Entry

Type choices, press Enter.
Option . . . . . : DL
Description. . . . . : DISPLAY LIBRARY LIST
Command. . . . . : DSPLIBL
```

```
F3=Exit  F4=Prompt   F12=Cancel
```

4. Specify the required parameters and press Enter.
5. Repeat the previous step for each option that you want to define.

Chapter 7. Verifying Batch Jobs

BYPASS2000 performs many activities in batch mode. It is important to check whether any problems occurred during batch processing, or whether BYPASS2000 needs additional information to complete a processing phase.

The following options are available from various BYPASS2000 menus:

- Display the conversion log. See “Displaying the Conversion Log” for information.
- Acknowledge requested information. See “Acknowledging Information Requests” on page 75 for information.

You can also use the following system commands:

- Work with submitted jobs (WRKSBMJOB)
- Work with all spooled files (WRKSPLF).

Displaying the Conversion Log

BYPASS2000 records any information about critical errors, analysis problems, missing sources, or other problems in the conversion log.

1. Type 22 on most BYPASS2000 menus to access the Display Conversion Log display.

```
                          Display Conversion Log

Type choices, press Enter.

  Conversion program. . . .  *ALL         *All, name, generic*
    Message type. . . . . .               W, I, E

  Source:
    Name  . . . . . .  . .  *ALL         *All, name, generic*
    Type  . . . . . .  . .  *ALL         *All, type
  Object:
    Name  . . . . . .  . .  *ALL         *All, name, generic*
    Type  . . . . . .  . .  *ALL         *All, type









F3=Exit         F12=Cancel
```

2. Select the messages in which you are interested. If you press Enter to accept the default settings, a list similar to the one below is displayed.

```
                         Display Conversion Log

                                              From date (YYMMDD) :
Type choices, press Enter.
  4=Delete    5=Display

Opt Program    Source     Type Message description
 _  ANPDATC030 I00197     PGM  Warning: Found complex CHGVAR expression at  ...
 _  ANPDATC030 I00230     PGM  Warning: Found complex CHGVAR expression at  ...
 _  UTLODCCR   SA67913    PGM  Program analysis ended because of a user inq ...
 _  ANPDAT010  CBLCOPY    PGM  Copy COPYMBR not found                       ...
 _  ANRDAT020  CPYCON     PGM  Duplicate key: source CPYCONS (CPY), I/O are ...
 _  ANRDAT020  CPYCON     PGM  Duplicate key: source CPYCONS (CPY), I/O are ...
 _  ANRDAT020  CPYCON     PGM  Duplicate key: source CPYCONS (CPY), I/O are ...
 _  ANRDAT020  CPYCON     PGM  Duplicate key: source CPYCONS (CPY), I/O are ...
 _  ANRDAT020  CPYCON     PGM  Duplicate key: source CPYCONS (CPY), I/O are ...
 _  ANRDAT020  CPYCON     PGM  Duplicate key: source CPYCONS (CPY), I/O are ...
 _  ANRDAT020  CPYCON     PGM  Duplicate key: source CPYCONS (CPY), I/O are ...
 _  ANRDAT020  CPYCON     PGM  Duplicate key: source CPYCONS (CPY), I/O are ...
 _  ANRDAT020  CPYCON     PGM  Duplicate key: source CPYCONT (CPY), I/O are  +

F3=Exit    F5=Refresh    F11=Display date,time,object       F12=Cancel
F13=Repeat option        F17=Top        F18=Bottom          F23=Clear log
```

Note: Conversion logs use a lot of space. Once you know that you no longer need a particular log, press F23 to clear it.

3. Type 5 to view the full message text in the Display Message Attributes display. Information about the source related to the message and the message description will appear.

   Note: If you need to report a problem to the Support Center, you should provide all the information from the Display Message Attributes display.

```
                       Display Message Attribute

Press Enter to continue.

Program . . . . . . . . . . : ANPDATC030

Source:
Name/Type . . . . . . . . . : I00197     PGM

Object:
Name/Type . . . . . . . . . : ANDATCST   FIL
Operation . . . . . . . . . : INSERT
Return code . . . . . . . . :

Message:
Message ID/Sequence/Type  . : BAP3001   08 W
Date/Time . . . . . . . . . :  5/06/98  10:26:03

Line Warning: Found complex CHGVAR expression at statement 0000006. BYP
     ASS2000 handled the first element only.

F3=Exit    F12=Cancel    F10=Display second-level message
```

4. Press F10 to display a second-level message.

```
:                                                                              :
: Second-level message text:                                                   :
:  The syntax is too complex for automatic conversion. Rewrite the statement  :
:   in a simplified way and run the analysis again. If the statement          :
:  cannot be simplified, you must convert it manually.                         :
:                                                                              :
:                                                                              :
:                                                                              :
:                                                                              :
:                                                                              :
:                                                                              :
:                                                                              :
:                                                                              :
:                                                                              :
:                                                                              :
:                                                                              :
:                                                                              :
:                                                                              :
:  F3=Exit     F12=Cancel                                                      :
:                                                                              :
:..............................................................................:
```

Acknowledging Information Requests

While BYPASS2000 performs activities in batch mode, it may need additional information about the application being processed. To find out whether there are any outstanding requests for information, perform the following:

1. Type 23 on any BYPASS2000 menu to access the Acknowledge Requested Information display.

```
                    Acknowledge Requested Information

Type choices, press Enter.

  Program:
    Name. . . . . . . . . .  *ALL         *ALL, name, generic*
    Type. . . . . . . . . .  *ALL         *ALL, PGM

  Phase  . . . . . . . . .   *            *=*ALL, A=Ana., P=Prop., C=Conv.

  Inquiry:
    Requester . . . . . . .  *ALL         *ALL, name, generic*
    Type. . . . . . . . . .  *ALL         *ALL, type

  Message type . . . . . .   *            *=*ALL, W, I, E




F3=Exit    F5=Refresh   F12=Cancel
```

2. Select the requests you want to look up. If you press Enter to accept the default settings, a list similar to the one below is displayed.

```
                      Acknowledge Requested Information

                                          Position to program. . .:
 Type choices, press Enter.
   4=Delete    5=Display

    Pgm. Program    Inq. Inquiry    Phs Chk Msg. Message
Opt Type Name       Type Requester          Type Description
 _  PGM  SA67913    CCR  UTLODCCR   A   N   W    Dynamic linking found. Use ...
 _  PGM  CPY1CAL    CCR  UTLODCCR   A   N   W    Dynamic linking found. Use ...









 F3=Exit      F5=Refresh      F12=Cancel      F15=Not acknow. only
 F17=Top      F18=Bottom      F23=Clear acknow. requests
```

3. Type 5 next to a message to see the full message text in the display. Information about the source that is related to the message and the message description will appear.
   If you press F10, a second-level message will display, if available.

```
                      Acknowledge Requested Information

 Program Name/Type . . . . . : SA67913    / PGM

 Requester phase . . . . . : A

 Check . . . . . : N

 Requester name. . . . . : UTLODCCR
 Type/Sequence number. . . . : CCR /     1

 Statement . . . . . :       0

 I/O-Area Name/Sequence Nbr. :                                /     0
 Field Name/Displacement . . :                                /     0

 Filename/Type . . . . . :                    /

 Message Type. . . . . : W
 Line Dynamic linking found. Use "Work with dynamic call" to assign valu
      es.

 F3=Exit   F12=Cancel   F21=Action   F24=Message acknowledged
```

4. You must provide the requested information, and confirm that the message has been acknowledged before BYPASS2000 can resume processing your application.

   When you press F21, the appropriate display for the type of requested information is shown. For example, for the message *Dynamic linking found*, pressing F21 opens the CALL Statement Management – Variable List display. If the type of requested information is UNK, F21 is disabled.

   Press F24 before proceeding.

Chapter 8. Loading User-Database Information

You must load user-database information so that BYPASS2000 can retrieve information about the file objects, and create an inventory that lists the physical files that are contained in the *xxxxxx*DAT library (or in the library you specified as the old object library for *FILE in your customized environment setup).

This process analyzes all the database files in the active conversion environment.

Note: It is important that you act on all messages that BYPASS2000 may issue during this and all subsequent processing phases. You should put your message queue in break mode, and only proceed to the next phase once all issues related to the current phase have been resolved.

1. Type 2 on the BYPASS2000 for AS/400 Main Menu to access the User-Database Information display.

```
                    User-Database Information (BPLODASDB)

Type choices, press Enter.

Environment library  . . . . . .   *LIBL         Name, *LIBL, *CURLIB
Run in batch . . . . . . . . . .   *YES          *YES, *NO

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

2. Press Enter to accept the default values and submit the job to batch.

   The environment library contains a number of parameters that BYPASS2000 uses. You do not need to specify a name for this library, because BYPASS2000 automatically adds this library to the library list, and its default is set to *LIBL. The object BPPARM that contains the parameter information is a data area (*DTAARA) that is located in the xxxxxxDB library.

Chapter 9. Analyzing Your Application

The analysis of your application that BYPASS2000 performs is called *memory-level analysis*. During this phase, BYPASS2000 analyzes all source statements that are related to memory definition and management, found in the following:

- Database objects
- COPY members (containing file descriptions)
- SQL source
- Program source.

BYPASS2000 then creates a repository of all storage areas (fields described by both databases and programs) of your application.

The various activities that are related to the analysis of your application are accessible from the BYPASS2000 Memory-Level Analysis display. Each analysis step must be completed before you can proceed to the next step.

```
BP4SANL               BYPASS2000 Memory-Level Analysis
                                                          System:   TORAS002
Select one of the following:

     1. Work with database information
     2. Analyze database information
     3. Work with COPY sources
     4. Analyze COPY sources
     5. Work with SQL sources
     6. Analyze SQL sources
     7. Work with program sources
     8. Analyze program sources

    10. Work with file overrides
    11. Work with dynamic calls

    13. Work with logical REDEFINES

                                                                   More...
Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
(C) COPYRIGHT HAL S.p.A. 1994, 1998.
```

Analyzing Databases

The analysis of your application's database is based on the inventory list of physical files that was created when you loaded user-database information, as described in "Chapter 8. Loading User-Database Information" on page 77.

While searching the appropriate source input library or file, BYPASS2000 will automatically link DDS source that has the same name as the physical file. You can perform the analysis either globally for all database files or for each file individually.

- To perform the analysis globally, type 2 in the BYPASS2000 Memory-Level Analysis menu to analyze database information.
- To perform the analysis individually, do the following:
  1. Type 1 on the BYPASS2000 Memory-Level Analysis menu to access the Work with Database Information - Select Parameters display:

```
                        Work with Database Information
                              Select Parameters


DDS  name . . . . : *ALL          *ALL, name, generic*



Application code  : *ALL          *ALL, name, generic*


Processing status           Analysis              Conversion

Not started   (0) :            Y                     Y
In progress (1-8) :            Y                     Y
Completed     (9) :            Y                     Y




F3=Exit   F12=Cancel
```

2. Select the parameters of your choice and press Enter.
3. In the Work with Database Information display type A for each file that you want to analyze.

```
                                Work with Database Information
                                                        Position to . . :
 Type choices, press Enter.
   2=Change    4=Delete    5=Display    8=Hold    9=Release
   I=Input source   O=Output source   V=View log  E=Reset cnv. flag K=Reset chk
   A=Analyze  D=Delete analysis  C=Convert   F=Override files    J=Dynamic call
Opt DDS  Name                              Ana     Cnv Gen Log Req Chk
 A  ADSDEMO                                 9       9
 _  ADU006P                                 9       9
 _  ADU007P                                 9       9
 _  ADU009P                                 9       9
 _  ADU011P                                 9       9
 _  ADU020P                                 9       9
 _  ADU049P                                 9       9
 _  CENDSP                                  9       9
 _  CENDSP1                                 9       9
 _  FPO                                     9       9
 _  GETB01LA                                9       9
 _  HHAVLX                                  9       9
 _  I00221                                  9       9
                                                                      More...
 F3=Exit F4=Prompt F5=Refresh F6=Create  F10=Reset flag F11=Toggle
 F12=Cancel F15=Include new member       F21=Command entry      F24=More keys
```

- After the analysis has been carried out, display the conversion log and check for any messages (option 22 from the BYPASS2000 Memory-Level Analysis menu).
- Acknowledge any requested information (option 23 in the BYPASS2000 Memory-Level Analysis menu) and take the appropriate action.

  Note: You must resolve any problems before entering the next analysis phase.

Analyzing COPYs

During this phase, BYPASS2000 analyzes the fields of the application, identifies their characteristics, and builds an image of each field that will later be used to identify date-related data.

The analysis of COPY source is based on the inventory list of sources. If you have set up a default conversion environment, the list of sources does not contain the COPY elements that are referred to by the programs and that are contained in the QCPYSRC and/or QLBLSRC file of the old source library*xxxxxx*OLD. Therefore, you must include them now.

Note: If you have set up a customized conversion environment, you can skip this step.

Loading COPYs

To include COPY members in the inventory list, do the following:

1. Type 3 on the BYPASS2000 Memory-Level Analysis menu to access the Work with COPYs display.
2. Press F15 to load the files.

Running the COPY Analysis

You can analyze COPY source either globally or individually.

To analyze all COPYs globally, type 4 in the BYPASS2000 Memory-Level Analysis menu.

```
                          Analyze COPYs (BPANLCPY)

Type choices, press Enter.

From COPY  . . . . . . . . . .   *FIRST        Name, *FIRST
To COPY  . . . . . . . . . . .   *LAST         Name, *ONLY, *LAST
Application code . . . . . . .   *ALL          Name, *ALL, *BLANK
Environment library  . . . . .   *LIBL         Name, *LIBL, *CURLIB
Run in batch . . . . . . . . .   *YES          *YES, *NO

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

To analyze individual COPYs, do the following:

1. Type 3 on the BYPASS2000 Memory-Level Analysis menu to access the Work with COPY Sources – Select Parameters display.

```
                              Work with COPYs
                             Select Parameters


COPY name . . . . : *ALL          *ALL, name, generic*

Source file . . . : *ALL          *ALL, name, generic*

Application code  : *ALL          *ALL, name, generic*


Processing status           Analysis              Conversion

Not started   (0) :            Y                     Y
In progress (1-8) :            Y                     Y
Completed     (9) :            Y                     Y




F3=Exit   F12=Cancel
```

Select the parameters of your choice and press Enter.

2. In the Work with COPYs display, type A for each file that you want to analyze.

```
                                Work with COPYs
                                                    Position to . . :
 Type choices, press Enter.
   2=Change    4=Delete    5=Display    8=Hold    9=Release
   I=Input source   O=Output source   V=View log  E=Reset cnv. flag K=Reset chk
   A=Analyze  D=Delete analysis  C=Convert   F=Override files    J=Dynamic call
Opt COPY Name  Type                       Ana     Cnv Gen Log Req Chk
 A  CPYCONS    RPG                         0       0
 _  CPYCONT    RPG                         0       0
 _  CPY00209   CBL                         0       0
 _  CPY00226   RPG                         0       0
 _  CPY1       CBL                         0       0
 _  CPY70077   RPG                         0       0
 _  CPY71330   RPG                         0       0
 _  GEPH09PA   CBL                         0       0
 _  GEPH10PA   CBL                         0       0
 _  GEZA15BA   CBL                         0       0
 _  LIKE1      RPG                         0       0
 _  MLF1       CBL                         0       0
 _  MULT1      RPG                         0       0
                                                                      More...
 F3=Exit F4=Prompt F5=Refresh F6=Create  F10=Reset flag F11=Toggle
 F12=Cancel F15=Include new member       F21=Command entry      F24=More keys
```

3. After the analysis has been carried out, display the conversion log and check for any messages (option 22 on the BYPASS2000 Memory-Level Analysis menu).

   The most common message at this stage indicates a missing application object, such as a file that was not found in the conversion environment. In this case, you must do the following:

   a. Add the object to the conversion environment (option 2, Load AS/400 Database Information, of the BYPASS2000 AS/400 Main Menu).
   b. Delete the existing memory-level analysis for all programs that use this file.
   c. Re-analyze the programs for which the message was generated.

4. Acknowledge any requested information (option 23 on the BYPASS2000 Memory-Level Analysis menu).

   Note: You must resolve any problems before entering the next analysis phase.

Analyzing SQL Source

The SQL sources that BYPASS2000 analyzes are CREATE TABLES and CREATE VIEWS.

The analysis of SQL source is based on the inventory list of sources. If you have set up a default conversion environment, the list of sources does not contain the SQL source located in the QQRYSRC file of the old source library *xxxxxx*OLD. Therefore, you must include them now.

Note: If you have set up a customized conversion environment, you can skip this step.

Loading SQL Source

To include SQL source in the inventory list, do the following:

1. Type 5 on the BYPASS2000 Memory-Level Analysis menu to access the Work with SQL display.
2. Press F15 to load the files.

Running the SQL Analysis

You can analyze SQL source either globally or individually.

To analyze all SQL source globally, type 6 in the BYPASS2000 Memory-Level Analysis menu to analyze SQL source.

```
                        Analyze SQL Sources (BPANLSQL)

Type choices, press Enter.

From source  . . . . . .   *FIRST        Name, *FIRST
To source  . . . . . . .   *LAST         Name, *ONLY, *LAST
Application code . . . . .   *ALL         Name, *ALL, *BLANK
Environment library  . . . . . .   *LIBL         Name, *LIBL, *CURLIB
Run in batch . . . . . .   *YES          *YES, *NO

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

To analyze individual SQL source, do the following:

1. Type 5 on the BYPASS2000 Memory-Level Analysis menu to access the Work with SQL - Select Parameters display.

```
                                Work with SQL
                              Select Parameters



Source name . . . : *ALL          *ALL, name, generic*

Application code  : *ALL          *ALL, name, generic*


Processing status     Analysis  Conversion

Not started   (0) :      Y          Y
In progress (1-8) :      Y          Y
Completed     (9) :      Y          Y




F3=Exit  F12=Cancel
```

2. Select the parameters of your choice and press Enter.
3. In the Work with SQL display, type `A` for each file that you want to analyze.

```
                                Work with SQL
                                                    Position to . . :
Type choices, press Enter.
  2=Change      4=Delete      5=Display      8=Hold      9=Release
  I=Input source    O=Output source      V=View log   E=Reset cnv. flag
  A=Analyze   D=Delete analysis   C=Convert   R=Show requested information
Opt SQL name     Source file  Library     Ana Cnv Log Req
 A  PROVATBL1    QQRYSRC      BP10OLD      0   0
 _  TXCBL14DB    QQRYSRC      BP10OLD      0   0
 _  TXCBL15DB    QQRYSRC      BP10OLD      0   0
 _  TXCBL16DB    QQRYSRC      BP10OLD      0   0









F3=Exit F4=Prompt F5=Refresh F6=Create F10=Reset conversion flag F11=Toggle
F12=Cancel  F15=Include new member     F21=Command entry         F24=More keys
```

4. After the analysis has been carried out, display the conversion log and check for any messages (option 22 on the BYPASS2000 Memory-Level Analysis menu).
5. Acknowledge any requested information (option 23 on the BYPASS2000 Memory-Level Analysis menu).

   Note: You must resolve any problems before continuing.

Analyzing Programs

During the program analysis, BYPASS2000 performs the following tasks:

- Obtain and store all program instructions in a repository file.
- Identify program fields and internal program areas.

- Build the caller/called relationship tree that will allow the passing and receiving of parameters.
- Build the relationship between variables in programs.
- Store information on fields that are not to be expanded because they belong to display or printer files.
- Check that all files, COPYs and tables referred to by a program have been loaded into the conversion environment, and flag any missing elements in the conversion log.
- Highlight fields that are used by the program but are not defined. This may be caused by incorrect versions of files or COPYs.

Loading Programs

The program analysis is based on the inventory list of sources. If you have set up a default conversion environment, the list of sources does not contain the source members of the source files that are located in the old source library *xxxxxx*OLD. You must include them now.

Tip: If you have set up a customized conversion environment, you can skip this step.

To include the source in the inventory list, do the following:

1. Type 7 on the BYPASS2000 Memory-Level Analysis menu to access the Work with Program display.
2. Press F15 to load the files.

Running the Program Analysis

You can analyze programs either globally or individually.

To analyze all programs globally, type 8 in the BYPASS2000 Memory-Level Analysis menu. BYPASS2000 first analyzes OCL and CLP programs, in order to store OVRDBF information in a repository file. Then it analyzes COBOL and RPG programs.

```
                        Analyze Programs (BPANLPGM)

Type choices, press Enter.

From program . . . . . .   *FIRST        Name, *FIRST
To program . . . . . .     *LAST         Name, *ONLY, *LAST
Application code . . . . . .  *ALL       Name, *ALL, *BLANK
Environment library  . . . . . .  *LIBL   Name, *LIBL, *CURLIB
Run in batch . . . . . .   *YES          *YES, *NO

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

To analyze individual programs, do the following:

1. Type 7 on the BYPASS2000 Memory-Level Analysis menu to access the Work with Program Sources– Select Parameters display.

```
                            Work with Programs
                            Select Parameters


Source name . . . : *ALL          *ALL, name, generic*

Source file . . . : *ALL          *ALL, name, generic*

Application code  : *ALL          *ALL, name, generic*


Processing status           Analysis  Propagation  Conversion

Not started   (0) :            Y           Y           Y
In progress (1-8) :            Y           Y           Y
Completed     (9) :            Y           Y           Y




F3=Exit   F12=Cancel
```

2. Select the parameters of your choice and press Enter.
3. In the Work with Programs display, type A for each file that you want to analyze.

```
                              Work with Programs
                                                  Position to . . :
 Type choices, press Enter.
   2=Change    4=Delete    5=Display    8=Hold    9=Release
   I=Input source      O=Output source  V=View log             A=Analyze
   D=Delete analysis   P=Propagate      Q=Delete propagation   C=Convert
Opt Pgm Name   Type                      Ana Prp Cnv Gen Log Req Chk
 A  ADU018R    RPG                        9   5   0
 _  ARRAY      RPG                        9   1   0
 _  CALLED     RPG                        9   0   0
 _  CALLER     RPG                        9   0   0
 _  CBLCOPY    CBL                        9   0   0
 _  CENDS      RPG                        9   0   0
 _  CENDS1     RPG                        9   0   0
 _  CENDS2     RPG                        9   0   0
 _  CENDS3     RPG                        9   0   0
 _  CPYCON     RPG                        9   0   0
 _  CPY1CAL    CBL                        9   0   0
 _  CPY1TST    CBL                        9   0   0
 _  DG0100     RPG                        9   0   0
                                                                        More...
 F3=Exit F4=Prompt F5=Refresh F6=Create  F10=Reset flag F11=Toggle F13=Repeat
 F12=Cancel  F21=Command entry  F23=More option      F24=More keys
```

4. After the analysis has been carried out, display the conversion log and check for any messages (option 22 from the BYPASS2000 Memory-Level Analysis menu).
5. Acknowledge any requested information (option 23 in the BYPASS2000 Memory-Level Analysis menu).

   Note: You must resolve any problems before continuing. To successfully complete the memory-level analysis of an application, you may need to add, remove, or change items in the conversion environment, as described in “Chapter 15. Changing the Conversion Environment” on page 139 .

   After adding items to the conversion environment, you must perform the memory-level analysis of the added items. If you use the global analysis options from the BYPASS2000 Memory-Level Analysis menu, BYPASS2000 will analyze only the newly added items that are indicated by the status flag set to ’0’.

Analyzing OCL Programs

During the analysis of OCL programs, BYPASS2000 simply obtains //FILE, //LOAD, and OVRDBF information and stores it in a repository file.

Working with File Overrides (OVRDBF)

A program may use the OVRDBF command to establish a link between an external filename and the corresponding internal filename that is used within a called program. This redefinition must be handled at the end of the program analysis.

Note: BYPASS2000 does not understand the OVRDBF command when it is specified within programs. However, BYPASS2000 may automatically find the OVRDBF during the analysis of CLP and OCL programs, if the overrides are static (but not dynamic).

1. Type `10` in the BYPASS2000 Memory-Level Analysis menu to access the Work with File Overrides display.

```
                    Work with File Overrides

                                                    Position to . . :
 Type choices, press Enter.
   2=Change     4=Delete    5=Display

Opt   Program Name  Filename     Target file









 F3=Exit     F5=Refresh     F6=Create
 F12=Cancel  F17=Top        F18=Bottom
 No data to list
```

2. Press F6 to add an entry for each program to establish a relationship among the following names:
   - The program for which the override is in effect
   - Its internal filename
   - Its target (real) filename.

```
                            Create New Entry



 Program name  . . :  TEST1

 Filename  . . . . :  MYFILE1

 Target file . . . :  MYFILEPF









 F3=Exit     F5=Refresh     F12=Cancel
```

Working with Dynamic Calls

BYPASS2000 uses the program names in CALL statements to create a link between the passed parameters of the caller, and the LINKAGE section or the entry parameter list of the called program. If the name of the called program is contained in a variable, BYPASS2000 tries to determine its correct value. If the name of the called program cannot be retrieved, BYPASS2000 requests user information.

1. Type 11 in the BYPASS2000 Memory-Level Analysis menu to access the Work with Dynamic Call Statements display.

You can establish an association between the variable that is used for the dynamic call and the name of the program that is being called.

```
                    Work with Dynamic CALL Statements
                              Variable list
 Pgm name: CPY1CAL
                                                        Position to . . :
 Type choices, press Enter.
   1=Select  4=Delete  5=Source lines  C=Check  *=Uncheck
   I=Original source         O=Converted source
                                                         Called    Chk Inf. CALL
Opt COPY Name  Stmt Nbr Program-Name Variable            Program       Pro. Type
 1                  36  FILE-NAME                        PGM2          PGM  CALL
 _                                                       PGM3          PGM  CALL
 _                  38  FILE-NAME                        PGM2          PGM  CALL
 _                                                       PGM3          PGM  CALL
 _
 F3=Exit           F12=Cancel                  F17=Top
 F18=Bottom        F23=Check all values        F24=Uncheck all values
```

2. Type 1 next to an entry and press Enter to access the Work with CALL Statements - Value List display.

```
                         Work with CALL Statements
                               Value list
 Pgm name: CPY1CAL

 CPY name:            Pgm Var: FILE-NAME                     Stmt Nbr:      36

 Called            Check
 Program Name      Flag
 PGM2
 PGM3
 F3=Exit   F5=Refresh   F6=Insert new name        F12=Cancel
 F17=Top   F18=Bottom   F23=Check listed values   F24=Uncheck listed values
```

3. Press F6 to create a new entry.

```
                          Work with CALL Statements
                               Insert new name
Type choices, press Enter.

Pgm name . . . . . :  CPY1CAL

CPY name . . . . . :

Pgm var. . . . . :  FILE-NAME

Stmt nbr . . . . . :       36

New Program Name . . . . . :  MYPGM









F3=Exit     F12=Cancel
```

Even if BYPASS2000 is able to determine the name of the called program, you may have to provide information. This is the case if during the conversion environment setup you have specified 'N' for the parameter ″Assume value of program-name variable was verified″, as described in “Choosing the Right Conversion Parameters” on page 8.

When BYPASS2000 requests user information is requested, the program analysis stops with the analysis flag set to 7. In this case, do the following:

1. Type 23 to acknowledge the requested information.
2. From the Acknowledge Requested Information display, press F21 to provide the requested information, and then press F24 to confirm the message.
3. Run the analysis again.

Note: When the flag is set to 7, you do not need to delete the result of the previous analysis before running the analysis again.

Working with Logical REDEFINES

When a COBOL program uses REDEFINES to remap an area of storage, BYPASS2000 attempts to determine if the redefined area refers to the same type of data in the area, or if it defines a completely different set of data.

- If BYPASS2000 thinks that the format of the data in the redefined area is similar enough to assume that the data contents are likely to be the same, then it considers both as a single I/O area.
- If BYPASS2000 determines that the redefined data seems to be of a completely different type, then it uses the same I/O area name for the second (and subsequent) definition, but differentiates them by a unique sequence number. BYPASS2000 thereby treats the redefined data as a separate piece of data.

Sometimes you may decide that BYPASS2000 has not made the appropriate choice for a specific use of REDEFINES. In that case, you may use option 13 (Work with Logical REDEFINES) to correct the way BYPASS2000 has handled the situation.

1. Type 13 in the BYPASS2000 Memory-Level Analysis menu to access the Select Program display.

```
                          Select Program
                                       Position to source type:
Type choice, press Enter.              Position to source name:
   1=Select

     Source
 Opt Type    Source Name  Source File  Library
  _  PGM     LIMIT        QRPGSRC      BP10OLD
  _  PGM     LIMIT6       QRPGSRC      BP10OLD
  _  PGM     LOD72340A    QRPGSRC      BP10OLD
  _  PGM     MLF001       QCBLSRC      BP10OLD
  _  PGM     PRTCALED     QRPGSRC      BP10OLD
  _  PGM     PRTCALLER    QRPGSRC      BP10OLD
  _  PGM     REPOUT       QS36SRC      BP10OLD
  _  PGM     RNP36        QS36PRC      BP10OLD
  _  PGM     RPGCOPY      QRPGSRC      BP10OLD
  _  PGM     SAIFOR       QCBLSRC      BP10OLD
  1  PGM     SAIFOR2      QCBLSRC      BP10OLD
  _  PGM     SAIFOR3      QCBLSRC      BP10OLD
  _  PGM     SANOTVAL     QCBLSRC      BP10OLD                              +

F3=Exit   F5=Refresh   F12=Cancel   F17=Top   F18=Bottom
```

2. Type 1 next to the program that contains the REDEFINES clause and press Enter to access the Select I/O Area display.

```
                          Select I/O Area
Pgm Name : SAIFOR2
                                 Position to I/O area:
Type choices, press Enter.
  1=Select

   I/O Area
Opt Name
 _  DATE-REC









F3=Exit      F5=Refresh      F12=Cancel      F17=Top      F18=Bottom
```

3. Type 1 next to the I/O area that you want to modify and press Enter to access the Display Field List display.

```
                          Display Field List
 Pgm name : SAIFOR2       I/O-area name: DATE-REC
                                              Position to Fld Displ:
Type choices, press Enter.
 4=Delete    5=Display full data    X,C,Y=Skip, force, avoid splitting

Opt S Source     L Field                                  Field Field F Fld Fld
  _  T Name       R Name                                  Displ Len.  T R C Grp
  _  P SAIFOR2      01  SCREENA                              1    38 X Y Y
  _  P SAIFOR2        05  SAIFORD-RECORD                     1    38 X
  _  P SAIFOR2        05  RECORD-I                           1    19 X Y Y  Y
  _  P SAIFOR2          06  RECORD-I-INDIC                   1     1 X Y    Y
  _  P SAIFOR2            07  IN03                          1     1 N Y
  _  P SAIFOR2          06  YMD1                            2     6 N Y
  _  P SAIFOR2          06  YMD2                            8     6 N Y
  _  P SAIFOR2          06  YMD3                           14     6 N Y
  _  P SAIFOR2    C   05  RECORD-O                           1    38 X Y Y  Y
  _  P SAIFOR2          06  YMD1                            1     6 N Y
  _  P SAIFOR2          06  YMD2                            7     6 N Y
  _  P SAIFOR2          06  YMD3                           13     6 N Y    +

 F3=Exit     F5=Refresh     F12=Cancel               F14=Set highlight rules
 F15=Only highlighted fields  F16=Repeat position to  F17=Top      F18=Bottom
```

4. Select one of the following actions for an area or field:
   - 'X' if BYPASS2000 considers the original and the redefined area to be two separate areas, and they should really be considered as a single area.
   - 'C' if BYPASS2000 has not separated the fields, when they should be separated into different areas.
   - 'Y' if you do not want a separate redefining area for the field.
5. Delete the program analysis for this program (option D in the Work with Program display)
6. Run the analysis for the program again (option A in the Work with Program display).

Working with Display and Print Areas

Use the Work with Display and Print Areas function when you want certain areas to behave like print or display areas. You can work with display and print areas in programs or COPYs.

To work with display and print areas in programs do the following:

1. Type 15 from the BYPASS2000 Memory-Level Analysis menu.
2. Specify the parameters of your choice in the Work with Program – Select Parameters display, and press Enter.
3. In the Work with Program display, type 1 next to a program.
4. Press F6 to insert a new field.

```
                          Insert New Field

Source Type. . . . PGM                          Type

Source Name. . . . ADU018R                      Name

Information type . PRT                          PRT, MAP


I/O-area attributes.
Name . . . . . . . *ALL                         *ALL(only for select), Name
Sequence . . . . .   0                          Number


Field attributes.
Name . . . . . . . .  MYFIELD                   Name
Displacement . . .     0                        Number
Length . . . . . .     6                        Number




F3=Exit   F4=Select field   F5=Refresh   F12=Cancel
```

5. Supply the required information for the new field:
6. Delete the program analysis (option D in the Work with Program display)
7. Run the analysis for the program again (option A in the Work with Program display).

To work with display and print areas in COPYs do the following:

1. Type 16 from the BYPASS2000 Memory-Level Analysis menu.
2. Specify the parameters of your choice in the Work with COPYs – Select Parameters display, and press Enter
3. In the Work with COPYs display, type 1 next to a COPY.
4. Press F6 to insert the new field.
5. Delete the COPY analysis (option D in the Work with COPYs display)
6. Run the analysis for the COPY again (option A in the Work with COPYs display).

Working with CALL-Parameter Types

Use the Work with CALL-Parameter Types display when you have programs that are linked by CALL statements and you want to specify a CALL parameter as a display or print area.

1. Type 17 from the BYPASS2000 Memory-Level Analysis menu to access the Work with CALL Parameters display.

```
                          Work with CALL Parameters

                                                      Position to . . :
 Type choices, press Enter.
   2=Change     4=Delete

Opt Pgm Name    Param. Nbr    Info Type
 _  PRTCALED      1           PRT









 F3=Exit     F5=Refresh     F6=Create
 F12=Cancel  F17=Top        F18=Bottom
```

2. Press F6 to create a record and enter the required information.

```
                             Create a Record

 Called program
 Program name  . . . .  MYPGM

 Parameter number. . .      1

 Info type  (MAP/PRT):    PRT









 F3=Exit     F5=Refresh     F12=Cancel
```

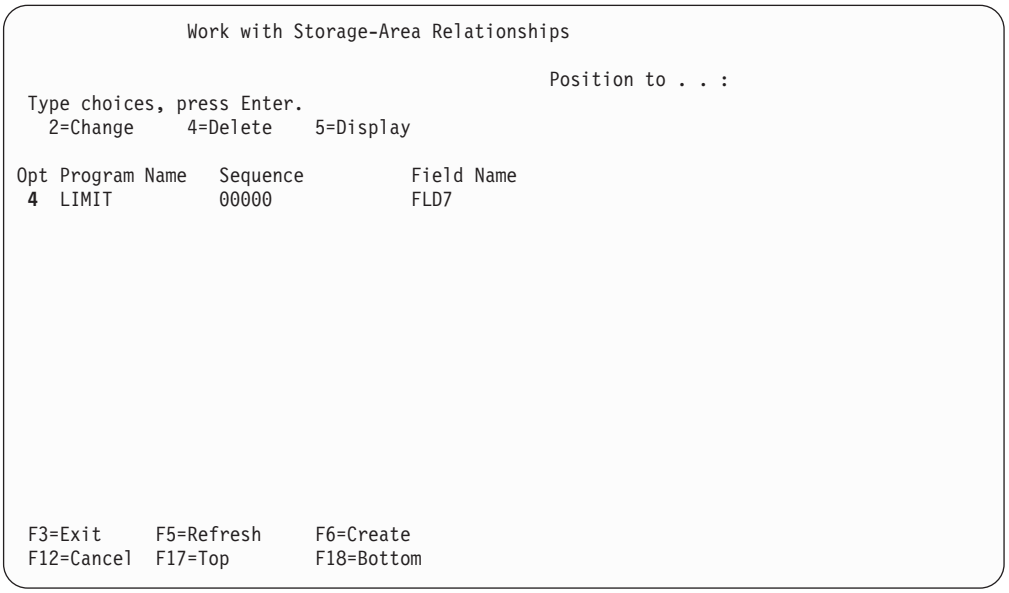
3. Delete the program analysis (option D in the Work with Program display)
4. Run the analysis for the program again (option A in the Work with Program display).

Deleting Relationships Between Storage Areas

Sometimes the propagation analysis finds a high number of generic fields that are caused by the use of general communication areas. If this is the case, you may want to disable the relationships between certain storage areas and thus disable propagation from and to specific areas. (See “Chapter 13. Propagating Date Fields” on page 123 for a detailed discussion of the propagation analysis phase.)

To disable all relationships between a specific generic storage area and every other field that is in contact with it, do the following:

1. Type 19 from the BYPASS2000 Memory-Level Analysis menu to access the Work with Storage-Area Relationships display.



2. Type 4 next to the program for which you want to delete the relationship with a field.

Deleting Analysis Results

You can delete the results of the memory-level analysis from the conversion environment, if necessary. You can perform the deletion in either batch or interactive mode, for all source or for subsets only.

Use the following options on the Memory-Level Analysis menu to delete different types of analysis results:

Option 40. Delete database analysis
Deletes the DDS analysis.

Option 41. Delete COPY analysis
Deletes the COPY analysis.

Option 42. Delete SQL analysis
Deletes the COPY analysis.

Option 43. Delete program analysis and propagation
Deletes the program analysis as well as any related propagation results.

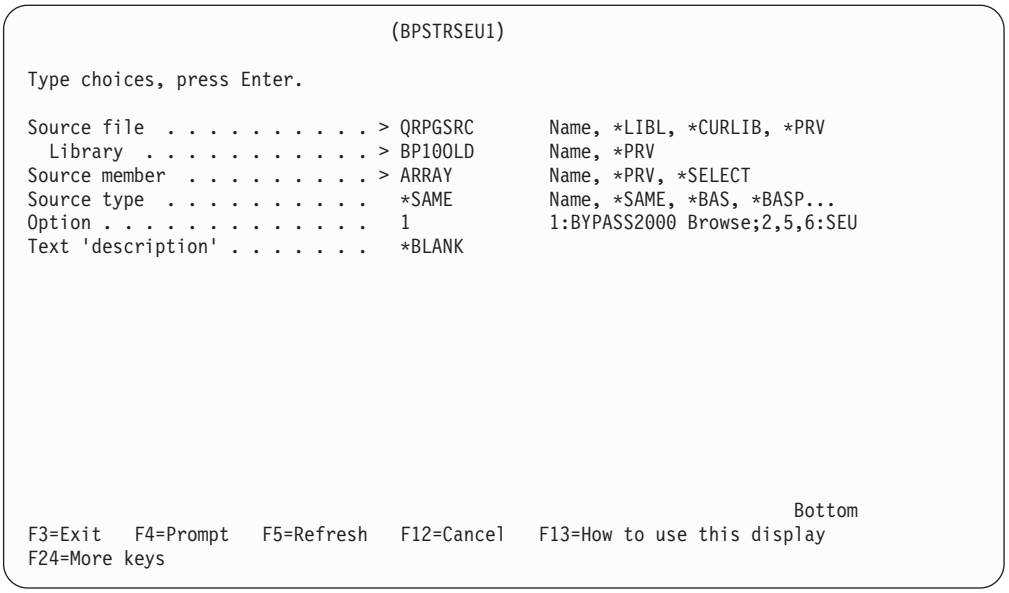
Option 44. Delete database, COPY, and program analysis and propagation
Deletes the database, COPY, and program analysis, as well as any related propagation results.

Chapter 10. Browsing Source Code

The BYPASS2000 Browser is an SEU-like browser. Use it to display date-field properties. The browser has a command-line interface for SEU-type commands, as well as a set of specialized function keys. From the Source Program display you can work in SEU browse mode. To enter a command, specify the corresponding option on the command line, or press one of the function keys.

To view source code in the BYPASS2000 Browser, do the following:

1. In most ″Work with ....″ displays, type `I` or `0` next to an entry.
2. Press F4 to access the selection screen.



```
                                (BPSTRSEU1)

Type choices, press Enter.

Source file  . . . . . . . . . . > QRPGSRC       Name, *LIBL, *CURLIB, *PRV
  Library  . . . . . . . . . . . > BP10OLD       Name, *PRV
Source member  . . . . . . . . . > ARRAY         Name, *PRV, *SELECT
Source type  . . . . . . . . . .   *SAME         Name, *SAME, *BAS, *BASP...
Option . . . . . . . . . . . . .   1             1:BYPASS2000 Browse;2,5,6:SEU
Text 'description' . . . . . . .   *BLANK

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

3. Specify one of the following options:

   Option 1
   Start the BYPASS2000 Browser. This is the default.

   Option 2
   Start SEU in Edit mode.

   Option 5
   Start SEU in Browse mode.

   Option 6
   Print the source.

   If you select option 1, the BYPASS2000 Browser displays source code.

Chapter 11. Assigning Date Fields

Up to this point, BYPASS2000 has performed its analysis on *all* the data in the files and programs of your application. The next step is to provide the application with the necessary date-related information, such as the fields which contain dates and their appropriate date format as well as any related fields which need to be widened to accommodate century information.

Once you have identified the date fields in your application database, BYPASS2000 will propagate this date information to the other fields in the programs that it has already identified. Identifying date fields is called *assigning*, *seeding*, or *anchoring*. These terms are used interchangeably.

Note: Assigning date fields correctly is most important for the successful conversion of your application. If you fail to assign a date field in the database, BYPASS2000 assumes that this field does not contain year-related information. Consequently, BYPASS2000 does not propagate the year-sensitive attribute from this field to related fields throughout your application. This may lead to an incomplete conversion of your application. Errors made while assigning date fields may subsequently lead to propagating incorrect fields.

You specify date fields and their formats in the BYPASS2000 Date-Field Assignment menu and its related displays. Type `4` in the BYPASS2000 for AS/400 Main Menu. The BYPASS2000 Date-Field Assignment menu appears.

```
BPSRCASS              BYPASS2000 Date-Field Assignment
                                                          System:   TORAS002
Select one of the following:

     1. Assign date field for I/O area related to file
     2. Assign date field
     3. Assign I/O area to related file
     4. Assign record type to related I/O area
     5. Work with migration utility programs
     6. Print list of assigned date fields
     7. Work with user-default date
     8. Work with date fields not to expand or propagate
     9. Assign dates from files to program and COPY areas
    10. Work with dates in files
    11. Import external field assignment (HSDATDFI)
    12. Import external field assignment (HSDATDFN)
    13. Load field assignment into HSDATDFN for export
    14. Create interface from IBM SEARCH2000 repository
                                                                    More...
Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
(C) COPYRIGHT HAL S.p.A. 1994, 1998.
```

This menu offers a number of options that you can use to perform the following tasks:

- “Assigning Fields and I/O Areas” on page 19.
- “Working with Dates” on page 22.
- “Importing or Exporting Date Information” on page 21

You can assign date fields anywhere; including fields that are defined in programs. BYPASS2000 only requires date assignments in the database files. This makes it easier and more efficient for the application to locate other fields.

Assigning Date Fields in Dictionaries

If your application uses dictionaries (field-reference files), you should assign the date fields in these dictionaries first, as described in “Assigning Date Fields in Database Files” on page 102.

Note: The assignment steps are the same for date fields in dictionaries as for date fields in files that are not dictionaries. However, after you have assigned all date fields in the dictionaries, you must spread this assignment from the dictionaries to the database (option A). Once you have spread the assignment from the dictionaries to all files that reference them, you can then perform the assignment for each file that is not a dictionary, but contains date fields.

1. Type 1 in the BYPASS2000 Date-Field Assignment menu to access the display that is shown below.

```
                Assign Date Field for I/O Area Related to File
                        Select Source Name or I/O Area

Type choices, press Enter.

  Source:
    Name  . . . . .  *ALL                          *All, name, generic*

  I/O area:
    Name  . . . . .  *ALL                          *All, name, generic*
    Sequence. . . .    0                           Number

  File:
    Type  . . . . .  *ALL                          *All, name
    Name  . . . . .  *ALL                          *All, name









F3=Exit  F12=Cancel
```

2. Specify the parameters of your choice and press Enter.

```
                              Assign Date Field
                                I/O-Area List
                                                    Position to . . :
 Type choices, press Enter.
   1=Select    6=Print assigned date fields   A=Assign from dict. to database
   2=Set/Reset file as dictionary                    ---- File attributes -----
Opt Src. Name  I/O-Area Name                  Seq A  Type  Name              D
 _  ADSDEMO    *DEMOR                         000    PF    ADSDEMO
 _  ADU006P    *AD006R                        000    PF    ADU006P
 _  ADU007P    *AD007R                        000    PF    ADU007P
 _  ADU009P    *AD009R                        000    PF    ADU009P
 _  ADU011P    *AD011R                        000    PF    ADU011P
 _  ADU020P    *AD020R                        000    PF    ADU020P
 _  ADU049P    *ADU49R                        000    PF    ADU049P
 _  CENDSP     *RENDSP                        000    PF    CENDSP
 _  CENDSP1    *RENDSP                        000    PF    CENDSP1
 _  CPY70077   SA                             000    PF    SA70077
 _  FPO        *FPOF1                         000    PF    FPO
 _  GETB01LA   *RGETB01LA                     000    PF    GETB01LA
 _  HHAVLX     *HHAVLR                        000    PF    HHAVLX            +

 F3=Exit     F5=Refresh     F12=Cancel
 F17=Top     F18=Bottom     F21=Print assigned date fields of listed sources
```

3. For each dictionary, type 2 to set the file as a dictionary.
4. For each dictionary, type 1 to assign date fields in the Assign Date Field – Field List display.

```
                         Assign Date Field - Field list
 DDS name  ADSDEMO     I/O area *DEMOR                       Seq   0
 Search field name:                                Position to displ:

Type choices, press Enter.
 D=Assign date field    1-7=Quick-assign date field   W=Work with assigned date
 *=Annul assignment   R=Reusable field                   V=View all attributes
    Data  Field                                F       F Field Fld  Field Int. D
Opt Type  Name                                 R  Occ. G Displ Type Len.  Dec. A
 _        01  DEMOR                                    Y     1 CHAR   119  0 0

 _          02  DMNETW                                      1 CHAR     2  0 0
              NETWORK
 _          02  DMNNRL                                      3 CHAR     1  0 0
              NETWORK NRL
 _          02  DMEFFT                                      4 PKD      4  7 0
              EFFECTIVE DATE-CYYMMD
 _          02  DMFPRJ                                      8 ZND      4  4 0
              FROM PROJECT YYQT                                              +


 F3=Exit F5=Refresh  F11=View colhdg F14=Set HI rules  F15=View only highlight
 F17=Top F18=Bottom  F19=Field type  F20=Default type  F21=PF contents F12=Canc.
```

5. To assign a field as a date field do one of the following:
   a. Type D next to the field name and press Enter.

      BYPASS2000 prompts you for additional information about this field, including its format, as described in “Providing Details about a Date Field” on page 104.

   b. Type an integer between 1 and 7 next to the field name to quick-assign the field as a date field. These integers correspond to the date types 001 through 007. (See “Date Types” on page xiv for a list of the supported date types.)
6. Spread the assignment from the dictionary to all databases (option 'A’ in the Assign Date Field – I/O Area List display).

7. Verify the field assignment in the database files.

To assign the fields of the dictionary, and automatically spread them to all the files that refer to the dictionary, do the following:

If necessary, you can assign the dictionary field with one date type, and then change the assignment to a different date type in one or more of the files that refer to the dictionary.

Note: If fields in files have been assigned from a dictionary, their assignment will not change if you modify the assignment in the dictionary and type A again.

Modifying Field Assignment in Dictionaries

To reassign fields in a dictionary for which you have already spread the ″year-sensitive″ attribute to the corresponding physical files, do the following:

1. Delete the previous assignments from all files that refer to the dictionary, as described in “Deleting the Assignment of a File Field” on page 115.
2. Reassign the date fields in the dictionary, as described in “Assigning Date Fields in Database Files”.
3. Repeat the assignment from the dictionary, as described in “Assigning Date Fields in Dictionaries” on page 100.
4. Delete the current propagation results, if any (option 42 on the BYPASS2000 Propagation-Level Analysis menu) and perform the propagation analysis again, as described in “Chapter 13. Propagating Date Fields” on page 123.

These steps are necessary, because assignment from the dictionary does not replace any existing field definitions.

Note: Modifying the assignment of dictionary (reference file) fields is very time consuming and should only be used for smaller applications. For medium-size and large applications you should examine whether it would be better to assign date fields in specific files rather than re-assigning date fields from the dictionary.

Assigning Additional Dictionary Fields

To assign additional fields in the dictionary (field reference file), do the following:

1. Assign the date fields in the dictionary, as described in “Assigning Date Fields in Database Files”.
2. Repeat the assignment from the dictionary, as described in “Assigning Date Fields in Dictionaries” on page 100.
3. Delete the current propagation results, if any (option 42 on the BYPASS2000 Propagation-Level Analysis menu) and perform the propagation analysis again, as described in “Chapter 13. Propagating Date Fields” on page 123.

Assigning Date Fields in Database Files

Assigning a field as a date field consists of two steps. The first step consists of identifying the field. The second step consists of providing information about the field.

Note: The following section assumes that your application uses external field descriptions from either DDS or SQL. If your application uses program-described files, see “Chapter 20. Assigning Date Fields in Program-Described Files” on page 171 before proceeding.

1. Type 1 in the BYPASS2000 Date-Field Assignment menu to access the Assign Date Field display.

```
               Assign Date Field for I/O Area Related to File
                       Select Source Name or I/O Area

 Type choices, press Enter.

   Source:
     Name  . . . . .  *ALL                          *All, name, generic*

   I/O area:
     Name  . . . . .  *ALL                          *All, name, generic*
     Sequence. . . .    0                           Number

   File:
     Type  . . . . .  *ALL                          *All, name
     Name  . . . . .  *ALL                          *All, name




 F3=Exit  F12=Cancel
```

2. Specify the parameters of your choice and press Enter.

```
                             Assign Date Field
                               I/O-Area List
                                                 Position to . . :
 Type choices, press Enter.
   1=Select    6=Print assigned date fields   A=Assign from dict. to database
   2=Set/Reset file as dictionary                    ---- File attributes -----
Opt Src. Name  I/O-Area Name                  Seq A  Type  Name               D
 _  ADSDEMO    *DEMOR                         000    PF    ADSDEMO
 _  ADU006P    *AD006R                        000    PF    ADU006P
 _  ADU007P    *AD007R                        000    PF    ADU007P
 _  ADU009P    *AD009R                        000    PF    ADU009P
 _  ADU011P    *AD011R                        000    PF    ADU011P
 _  ADU020P    *AD020R                        000    PF    ADU020P
 _  ADU049P    *ADU49R                        000    PF    ADU049P
 _  CENDSP     *RENDSP                        000    PF    CENDSP
 _  CENDSP1    *RENDSP                        000    PF    CENDSP1
 _  CPY70077   SA                             000    PF    SA70077
 _  FPO        *FPOF1                         000    PF    FPO
 _  GETB01LA   *RGETB01LA                     000    PF    GETB01LA
 _  HHAVLX     *HHAVLR                        000    PF    HHAVLX             +

 F3=Exit     F5=Refresh     F12=Cancel
 F17=Top     F18=Bottom     F21=Print assigned date fields of listed sources
```

3. Type 1 next to a sources and press Enter.

```
                          Assign Date Field - Field list
 DDS name  ADSDEMO     I/O area *DEMOR                        Seq   0
 Search field name:                                  Position to displ:

Type choices, press Enter.
 D=Assign date field    1-7=Quick-assign date field   W=Work with assigned date
 *=Annul assignment   R=Reusable field                   V=View all attributes
    Data  Field                                F       F Field Fld  Field Int. D
Opt Type  Name                                 R  Occ. G Displ Type Len.  Dec. A
 _        01  DEMOR                                    Y     1 CHAR   119  0 0

 _          02  DMNETW                                      1 CHAR     2  0 0
              NETWORK
 _          02  DMNNRL                                      3 CHAR     1  0 0
              NETWORK NRL
 _          02  DMEFFT                                      4 PKD      4  7 0
              EFFECTIVE DATE-CYYMMD
 _          02  DMFPRJ                                      8 ZND      4  4 0
              FROM PROJECT YYQT                                              +


 F3=Exit F5=Refresh  F11=View colhdg F14=Set HI rules  F15=View only highlight
 F17=Top F18=Bottom  F19=Field type  F20=Default type  F21=PF contents F12=Canc.
```

4. To assign a field as a date field, type `D` next to the field name and press Enter. BYPASS2000 prompts you for additional information about this field, including its format, as described in “Providing Details about a Date Field”.

   Tip: You can also quick-assign the field, as described in “Quick-Assigning a Date Field” on page 105.

5. Check all other files to find dates that have been overlooked, or to verify their format. Press F21 to see the data in each file.

Providing Details about a Date Field

Once you assign a date field with `D` in the Assign Date Field – Field List display, the Create New Date Field Entry display appears prompting you for information about the field. See “Rules for Assigning Date Fields” on page 19 for additional information.

```
                      Create New Date Field Entry

 Type choices, press Enter.

 Source name . . . . . :  ADSDEMO
 I/O-area name . . . . . :  *DEMOR
 I/O-area sequence number. . . :  000
 Field name. . . . . :  DMEFFT
 Displacement. . . . . :  00004
 Type. . . . . :  PKD
 Length. . . . . :  00004
 Integer / Decimal . . . . . :   7   0

 Date type . . . . . :  002  YEAR,MONTH,DAY
 Position-field substring. . . :  00001
 Length-field substring. . . . :  00004
 Field type. . . . . :  P
 Year length . . . . . :  2
 Expansion type. . . . . :  0   (0=Default /1=Not expand /2=Expand + one
                                reusing /3=Expand + one /4=Expand  + two)
 Propagation type. . . . . :  0   (0=Propag./1=Not propag./2=Upon confirm.)
 F3=Exit    F4=List date type    F5=Refresh    F12=Cancel
```

BYPASS2000 automatically determines a possible date type for the field. It is important that you verify this date type and make any appropriate changes.

1. If the correct date type is prefilled, press Enter.

   Tip: You can accept the default values for the remaining parameters.

2. If you need to change the date type for this field, do the following:
   a. Press F4 for a list of formats that BYPASS2000 supports.

      In the Display Date Type screen, scroll down to see more date types. If the display does not identify any additional date types, see “Date Types” on page xiv for a list of supported date types.

   b. Type 1 next to the correct format for the field, and press Enter.

      Note: BYPASS2000 only needs to know the position of the year in a date. The month and day positions are not significant. Therefore, date type 003 is valid for both DMY and MDY, even though its abbreviation only specifies DMY.

   c. You may accept the default values for the remaining parameters.

      By default, BYPASS2000 propagates date fields. The expansion default (expansion type 0) depends on the conversion style that you have specified on the **Default conversion style** parameter, when you created the conversion environment. See “Choosing the Right Conversion Style for Your Application” on page 5 for a list of valid conversion styles.

      You can override the defaults for individual fields by specifying any of the following settings:

      Expansion type 0
      Use the default expansion type for the chosen conversion style.

      Expansion type 1
      Do not expand the field.

      Expansion type 2
      Expand the field by 1 digit, reusing the half-byte that is available in packed fields with an odd number of digits. If you cannot reuse the half-byte, select expansion type 3.

      Expansion type 3
      Expand the field by 1 digit to add the century flag.

      Expansion type 4
      Expand the field by 2 digits to add the full century.

      See “Specifying Date-Field Expansion and Propagation” on page 20 for a discussion of the different values for the Expansion type and Propagation type parameters.

   d. Press Enter to assign the field.

Quick-Assigning a Date Field

Type an integer between 1 and 7 next to the field name to quick-assign the field as a date field. These integers correspond to the date types 001 through 007. This approach is faster than specifying option ’D’ to assign a field, because you do not need to provide any additional information in the Create New Date Field Entry display.

See “Date Types” on page xiv for a list of the supported date types.

Reusing a Field

Option R in the Assign Date Field - Field list display offers an alternative to expanding a field. You can actually use an alphanumeric field, or part of a field in a record or a data structure, to insert the century information of a date field without expanding the record or the data structure.

Consider the following example:

A record of length 8 is divided into FIELDA and FIELDB with the following DDS:

```
******** Beginning of data ************************************
        A          R TESTR                     TEXT('DATE')
        A
        A            FIELDA         2A
        A            FIELDB         6A
****************** End of data ********************************
```

- FIELDA is a year-sensitive alphanumeric field of length 2 (date type 001).
- FILEDB is an alphanumeric field of length 6.

Suppose that FIELDA (Year) is the only date field present in this file. If you assign FIELDA as date type 001, expansion type 0, and propagation type 0, the result after the conversion will be the following:

- The record length becomes 10.
- FIELDA is expanded from 2 digits to 4, which accounts for the overall increase of the record length.
- FIELDB remains an alphanumeric field of length 6.

Instead, you can do the following:

1. Specify option R for FIELDB, and specify that the new length of the field is 4.

   BYPASS2000 will reuse the last 2 digits of FIELDB to insert the century information for FIELDA.
2. Assign FIELDA as date type 001, expansion type 0, and propagation type 0, as before.

The result after the conversion will be the following:

- The overall record length remains 8 digits.
- FIELDA is expanded from 2 to 4 digits, through the reuse of 2 digits from FIELDB.
- FIELDB becomes an alphanumeric field of length 4.

Removing the Assignment of a Field

To remove the assignment of a field, simply type * next to the field in the Assign Date Field - Field list display, and press Enter.

Modifying an Assigned Field

To make changes to an assigned field, do the following:

1. Type W next to the field in the Assign Date Field - Field list display, and press Enter. The Work with Assigned Date Field display appears.

```
                        Work with Assigned Date Field
PGM name  ADSDEMO     I/O area *DEMOR                        Seq   0
Field name : DMEFFT                              Displacement . . :     4
Field type : PKD         Length . . :     4      Integer / Decimal:  7  0

Type choices, press F3 to confirm your update.
2=Change       *=Annul assignment

Opt Dtf  Description---------  Date Pos  Position  Length  Type Year Sty  Pro
 _  001  YEAR                 00006     00001     00004    P    2    0    0









F3=Exit with Confirm        F5=Refresh       F6=Create new date field
F12=Cancel     F17=Top      F18=Bottom
```

Figure 16. Work with Assigned Date Field

2. Type 2 next to an entry to access the Create New Date Field Entry display.

```
                     Create New Date Field Entry

Type choices, press Enter.

Source name . . . . . :  ADSDEMO
I/O-area name . . . . . :  *DEMOR
I/O-area sequence number. . . :  000
Field name. . . . . :  DMEFFT
Displacement. . . . . :  00004
Type. . . . . :  PKD
Length. . . . . :  00004
Integer / Decimal . . . . . :   7   0

Date type . . . . . :  001  YEAR
Position-field substring. . . :  00001
Length-field substring. . . . :  00004   Date position . . . :   00006
Field type. . . . . :  P
Year length . . . . . :  2
Expansion type. . . . . :  0   (0=Default /1=Not expand /2=Expand + one
                               reusing /3=Expand + one /4=Expand  + two)
Propagation type. . . . . :  0   (0=Propag./1=Not propag./2=Upon confirm.)
F3=Exit    F4=List date type    F5=Refresh    F12=Cancel
```

3. Make any required changes and press Enter twice to return to the Assign Date Field - Field list display.

Displaying File Data

Press F21 in the Assign Date Field - Field List display to show the data in the file and see the name, length, and type of each field. This is useful to verify if the field is actually a date field, and what format it is in.

Note: You cannot assign fields that contain decimal positions or that have a length of 1.

```
                              Display Report
                                          Report width . . . . . :      84
Position to line  . . . . .  _______    Shift to column  . . . . . .   ____
Line   ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
       PRDNBR  PRDDES                PRDEXP      PRDPRC   PRDQTA   SPLNBR  PRD
000001 00001   Marker                73,096        2.50       10   00001    00
000002 00002   Eraser (10 pieces)    73,097        5.65       63   00001    00
000003 00003   Hilighter - Red (10)  73,096        6.00      131   00003    00
000004 00004   Fountain Pen          73,097       10.50        9   00002    00
000005 00005   Ball Pen (10 pieces)  73,096        7.00      396   00003    00
000006 00006   Mini Stapler          73,099        7.00       37   00001    00
000007 00008   Binder Clips          73,096        4.00    1,000   00004    00
000008 44551   Notepad               73,098        3.50      500   00001    00
000009 66111   Holes Machine         73,098       13.50       50   00001    00
****** ********  End of report  ********
                                                                      Bottom
F3=Exit      F12=Cancel      F19=Left      F20=Right      F21=Split
```

Note: Date fields are not formatted in this query. For example, in the sample display above, the PRDEXP column (for the product expiration date) shows the entry 73,096 without leading zeros. You should interpret this entry as 07/30/96, which corresponds to July 30, 96, and indicates a MM/DD/YY format of the PRDEXP field.

Type V next to any field to see a list of all field attributes, as shown in the sample display below:

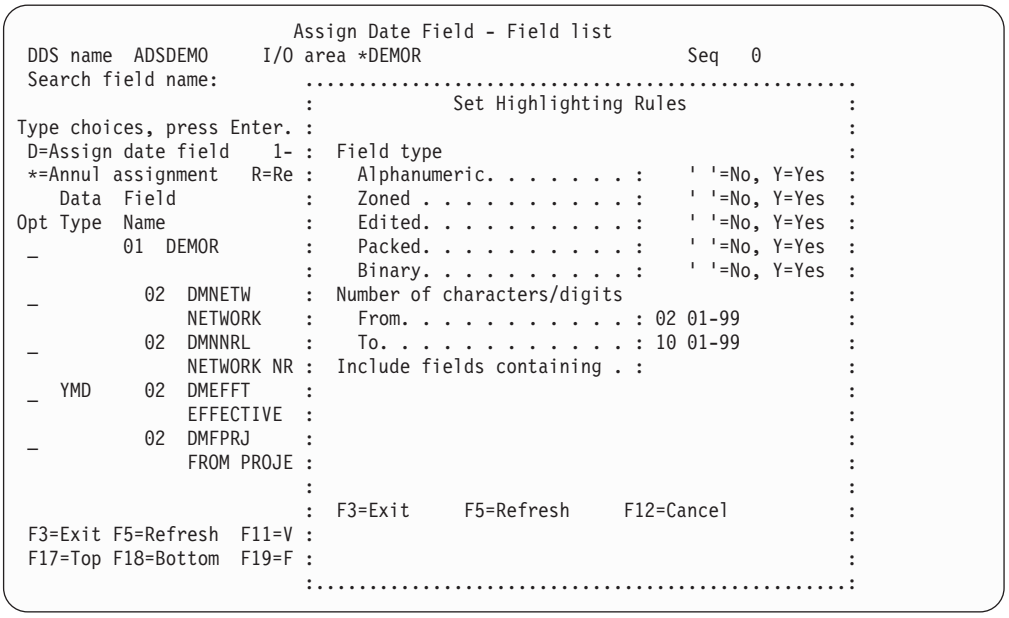
```
                        Display all field attributes

Source type : PGM   Source name : CHKDATC     Stmt Number :       4  Sect.: D

I/O area attributes
 Name . . . . : DATIN                      Seq.    0            Nmbr.     0

Field attributes
 Name . . . . : DATIN                      Disp.     1          Nmbr.     1
 Level  . . . : 01            Group field                 Type PKD
 Length . . . :     4         Integer/dec.    6 /   0     Sign     Just.

OCCURS and REDEFINES field attributes
 OCCURS . . . :     0         Total OCCURS      0
 OCCURS level :     0         OCCURS seq.       0         Shift     0
 REDEFINES  . :               REDEFINES nbr     0         Clause

Date field and new attributes
 Date type  . :   3    DMY   Day,Month,Year/Month,Day,Year
 Year length  : 2
 New displ. . :     1         New length        5

F3=Exit  F12=Cancel
```

Setting and Applying Highlighting Rules

You may not be interested in details about *all* date fields in the Assign Date Field - Field List display. To view only a subset that satisfies particular rules about name,

length, and type, you can use the function keys F14 and F15 to specify and highlight the fields in which you are most interested.

1. Press F14 to set highlighting rules, and to highlight the fields that satisfy these rules.



2. Press F15 to display only the subset of fields that satisfy the specified highlighting rules.

Assigning a Work Field That Contains Multiple Date Formats

Note: You cannot assign multiple date formats to a database field.

To assign a work field that is declared in COPY or program source and contains more than one date format, do the following:

1. Access the Assign Date Field – Field List display (as described in “Assigning Date Fields in Database Files” on page 102) and create an entry for the first date format.
2. Type `W` in the Assign Date Field – Field List display to access the Work with Assigned Date Field display.
3. Press `F6` to access the Create New Date Field Entry display, and create an entry for the second date format.
4. Repeat this step for each date format contained in the work field.

   Note: You can only assign the year in the positions in which it appears in the work field.

A work field that contains multiple date formats cannot be expanded or propagated. Valid values in the Create New Date Field Entry display are:

- Expansion type: 1
- Propagation type: 1 or 2

  Type 2 is preferred, as described in “Propagating Date Fields upon Confirmation” on page 25.

Tip: If your field has a length of 6 digits, and its format can be either DMY or YMD, you can use option 'M' to quickly assign the field with multiple date formats.

Assigning a Date Field That Is Used in Multiple Programs

If a field is used as a date field in multiple programs, you can assign it once for all programs. You can also lock the same field in a program that uses it as a non-date field.

1. Type 7 on the BYPASS2000 Date-Field Assignment menu to access the Work with User-Default Date display.

```
                        Work with User-Default Date

                                  Position to date field:
Type choices, press Enter.
 2=Change    4=Delete     5=Display

Opt I/O-Area Name             Field Name                Environment Language
```

```
 F3=Exit           F5=Refresh    F6=Create          F12=Cancel   F17=Top
 F15=Load default system date    F19=Work with add. parameters   F18=Bottom
 No data to list
```

2. Press F6 to access the Create New Entry display.

```
                           Create New Entry

 Environment  . . . . *ALL       (TRK IN PARAM.TABLE = BPENV)
 Language . . . . . . *ALL       (TRK IN PARAM.TABLE = BPLANG)
 I/O-area name  . . . *ALL

 Field attributes
 Name . . . . . . . .
 Displacement . . . .     0
 Length . . . . . . .     0                          Int.    0  Dec.   0
 Type . . . . . . . . X CHAR       ( /X/P/N/Z/B)     Sign    (Y/ )
 Level  . . . . . . .  0                             Group   (Y/ )

 Date attributes
 Position . . . . . .     1
 Length . . . . . . .     2
 Date type  . . . . .   0
 Year length. . . . . 2
 Expansion style  . . 0   (0=Default 1=Not expand 2=Expand + 1 (packed
                          fields) 3=Expand +1 4=Expand +2)
 Propagation type . . 0   (0=Propag. 1=Not propag. 2=Upon confirmation)
 F3=Exit          F4=List of environments/languages    F5=Refresh
 F12=Cancel       F19=Work with add. parameters
```

Specify the assignment and create the new field.

Locking the Field in Another Program

To lock the same field in another program that uses it as a non-date field, do the following:

1. Type 2 in the BYPASS2000 Date-Field Assignment menu to assign the date field.
2. Specify the parameters of your choice in the Assign Date Field – Select Source - I/O Area display and press Enter.
3. Type 1 on the field that you want to lock.
4. Do one of the following:
   - Type 2 on the field that you want to lock and press F3 to exit.
   - Type 1 on the field that you want to lock and press Enter, then type L next to the field, and press F3 to exit.

Assigning Only the Year Portion of a Date Field

If a field is not binary, you can assign any date by assigning the substring that corresponds to the date portion or the year value. If a field is packed, the substring that corresponds to the year value can only be 2 or 4 digits in length.

Assigning a Century or Century-Flag Field

If you want to assign a century or a century-flag field, you must first assign a field to the right of this field. The field must be assigned with the year on the left. As a result, the century or the century flag is positioned immediately to the left of the year, without any digits in between.

Note: If you reverse the order of these two steps and assign the century or century-flag field first, BYPASS2000 issues the following message: ″You cannot assign century data type if it is not followed by an assigned year field″.

If you try to delete the year assignment before deleting the century assignment, BYPASS2000 issues the following message: ″You cannot change or delete year date type, because century date type was found″.

Assigning Fields That Are Not Related to the Database

1. To assign fields that are dates but are described in areas not related to the database, access the Assign date field display (option 2 of the BYPASS2000 Date-Field Assignment menu).

```
                          Assign Date Field
                       Select Source - I/O Area

Type choices, press Enter.

  Source:
    Name  . . . . .  *ALL                          *All, name, generic*
    Type. . . . . .  CPY                           PGM, CPY

  I/O area:
    Name  . . . . .  *ALL                          *All, name, generic*
    Sequence. . . .    0                           Number









F3=Exit  F12=Cancel
```

BYPASS2000 automatically locks all database fields. As a result, it assumes that every field that is not assigned date information will never contain a date. The same does not apply to other areas. When assigning fields other than database fields, you must specify which part of a data area should be assigned date information.

2. Specify `PGM` as the source type, and press Enter to access the Assign Date Field – Select Source display.

```
                          Assign Date Field
                            Select Source
Type choices, press Enter.                      Position to . . :
  1=Select
                                                              Assigned
 Opt Program name Source file  Library      Analy. Prop.  Conv. Date

  1  ARRAY        QRPGSRC      BP10OLD        9      0      0
  _  CALLED       QRPGSRC      BP10OLD        9      0      0
  _  CALLER       QRPGSRC      BP10OLD        9      0      0
  _  CBLCOPY      QCBLSRC      BP10OLD        9      0      0
  _  CENDS        QRPGSRC      BP10OLD        9      0      0
  _  CENDS1       QRPGSRC      BP10OLD        9      0      0
  _  CENDS2       QRPGSRC      BP10OLD        9      0      0
  _  CENDS3       QRPGSRC      BP10OLD        9      0      0
  _  CPYCON       QRPGSRC      BP10OLD        9      0      0
  _  CPY1TST      QCBLSRC      BP10OLD        9      0      0
  _  DG0100       QRPGSRC      BP10OLD        9      0      0
  _  IFRPTP       QS36SRC      BP10OLD        9      0      0
  _  I00197       QCLSRC       BP10OLD        9      0      0               +

F3=Exit      F5=Refresh      F12=Cancel
F17=Top      F18=Bottom
```

3. Type `1` to access the Assign Date Field – Program I/O-Area List.

```
                  Assign Date Field - Program I/O-Area List
 Pgm name  ARRAY
                                Position to . . :
 Type choices, press Enter.
  1=Select   2=Select with lock    *=Annul all assignments and locks
  6=Print assigned dates                      -Assigned-
Opt I/O-Area Name                   Seq       Date  Lock
 1  AR                              000




 F3=Exit     F5=Refresh     F12=Cancel
 F17=Top     F18=Bottom     F21=Print assigned date fields of listed sources
```

4. To access the Assign Date Field – Field List, type 1 in the option column next to the field you want to assign and press Enter.

   If you want to assign a single field in an I/O area that contains multiple fields, you can either:

   - Type 1, if you want to assign the field, and leave the other fields open for propagation.
   - Type 2 if you want to assign the field, and lock all other fields in the same I/O area. This is the case when one field in the I/O area is a date field, while all the others do not contain year-related information.

```
                         Assign Date Field - Field List
 PGM name  ARRAY       I/O area AR                          Seq   0
 Search field name:                                Position to displ:

Type choices, press Enter.
 L=Lock area   A=Assign date fields, locking selected area level   5=Displ.attr.
 *=Annul lock  O=Assign date fields, locking only assigned fields
    Lock  Field                              R       G Field Fld  Field Int. D
Opt Type  Name                               F  Occ. F Displ Type Len.  Dec  A
 _        01  AR                                9999       1 ZND      6  6 0




 F3=Exit  F5=Refresh  F12=Cancel  F14=Set highlight rules F15=Only highlight
 F17=Top  F18=Bottom  F19=Work with field type  F20=Work with default type
```

5. Do one of the following:

   - Lock and assign an entire data structure (option 'A'=all)

     Use this option if at least one field is a date field. All fields are locked, and only the assigned field are considered to be date fields.
   - Assign only a particular field in a data structure (option 'O'=only)

Use this option if you do not have any information about other fields that will be assigned during propagation. Only one field is locked and assigned.

- Lock a data structure or a single field (option 'L'=lock).

  BYPASS2000 will consider this structure or field not to contain a date, unless you define a date within it. Use this option if the structure or field does not contain year-related information.

Assigning Fields in Printer and Display Files

By default, BYPASS2000 does not expand date fields in printer and display files. To override this default, do the following:

1. Type 8 in the BYPASS2000 Date-Field Assignment menu to work with fields that you do not want to be expanded.
2. Type 1 next to the program that contains the field that you want to expand. The Work with Field Not to Expand or Propagate display appears.

```
             Work with Not Expanded or Propagated Fields
Program name : SA72195
                                          Position to date:
 Type choices, press Enter.
 4=Delete  *=Undelete  5=Display full data
               Inf I/O Area    I/O                              Field Field Usr
Opt COPY Name  Typ Name        Seq.    Field Name               Displ Len.  Act
 _             MAP DISP-REC      0     DISP-REC                     1     0
 F3=Exit          F6=Insert new field         F12=Cancel
 F17=Top          F18=Bottom                  F23=Delete all user fields
```

3. Type 4 next to the field that you want to expand. This removes the field from the list of fields that are not to be expanded.

Modifying the Assignment of Fields

To modify the assignment of a field, do the following:

1. Delete the previous assignment for the field from all files, as described in "Deleting the Assignment of a File Field" on page 115.
2. Reassign the field, as described in "Assigning Date Fields in Database Files" on page 102 .

Deleting the Assignment of a File Field

It is not possible to perform a mass deletion of assigned file fields; you must delete the assignment for each field individually.

To delete the assignment of a file field, do the following:

1. Type 4 on the BYPASS2000 for AS/400 Main Menu to access the BYPASS2000 Date-Field Assignment menu.
2. Type 1 to access the Assign Date Field for I/O Area Related to File – Select Source Name or I/O Area display, if the incorrect field assignment occurred in a database.
3. Press Enter to list the selected sources in the Assign Date Field – I/O Area List display.
4. Type 1 to access the Assign Date Field – Field List display for the first file that contains date fields.

```
                        Assign Date Field - Field list
 COPY name CPY70077    I/O area SA                            Seq   0
 Search field name:                                  Position to displ:

Type choices, press Enter.
 D=Assign date field    1-7=Quick-assign date field   W=Work with assigned date
 *=Annul assignment   R=Reusable field                   V=View all attributes
    Data  Field                                F       F Field Fld  Field Int. D
Opt Type  Name                                 R  Occ. G Displ Type Len.  Dec. A
 _        01  SA                                            1 CHAR    10  0 0

 *  DMY     02  SA-DMY                                       1 ZND      6  6 0 Y

 *  MY      02  SA-MY                                        7 ZND      4  4 0 Y









 F3=Exit F5=Refresh  F11=View colhdg F14=Set HI rules  F15=View only highlight
 F17=Top F18=Bottom  F19=Field type  F20=Default type  F21=PF contents F12=Canc.
```

   BYPASS2000 presents a list of fields in this file. Fields that you have previously assigned are identified by an entry in the Data Type column, and a 'Y' in the Date Assigned column.
5. To delete the assignment of a field, type an asterisk (*) in the Option column, and press Enter.

Locking Fields

A field is automatically locked when you assign it. The lock function shows that you have made a decision about the year-sensitive nature of a field:

- If you only lock the field, this means that the field is not year-sensitive.
- If you lock and assign the field, this means that the field is year-sensitive.

Note: You must decide for each database field whether it is year-sensitive or not. Any field you assign is considered to be a date field. Any field you do not assign is considered not to be a date field and is automatically locked. If you fail to assign any date fields now, these fields will not be propagated. If by

mistake, you assign fields that are not date fields, they will be propagated. As a result of these omissions or errors, the propagation may be incomplete and/or incorrect.

To lock a field but not assign it, the following options are available:

- Database fields that are not year-sensitive are automatically locked by BYPASS2000. You do not need to lock these fields manually.
- Program fields that are not year-sensitive can be locked manually as follows:
  1. Type 2 in the BYPASS2000 Date-Field Assignment menu to assign date fields.
  2. Specify source type PGM in the Assign Date Field – Select Source - I/O Area display.
  3. Select the program that contains the field you want to lock (option 1).
  4. To lock an entire I/O area, type 2 and press F3 to exit the display.
  5. To lock a field in an I/O area without locking all other fields in the same I/O area, do the following:
     a. Type 0 to select the I/O area.
     b. Type 0 to select the field.
     c. Type L to lock the field and then press Enter.
     d. Press F3 to exit the display.

Chapter 12. Importing Date-Field Information

Instead of assigning date fields manually, you can import date-field information from an interface file that has a specific format. The interface file can be generated automatically by products such as IBM SEARCH2000, you can build it yourself, or BYPASS2000 can build it directly from a SEARCH2000 environment, as described in “Creating HSDATDFN from the SEARCH2000 Repository” on page 120.

Note: You cannot use external date-field information to assign the fields in internally-described files. You must assign these fields manually.

You can reuse the date field assignment from one environment in another one. In this case, BYPASS2000 can automatically generate the file HSDATDFN for you, as described in “Creating HSDATDFN from an Existing BYPASS2000 Environment” on page 119.

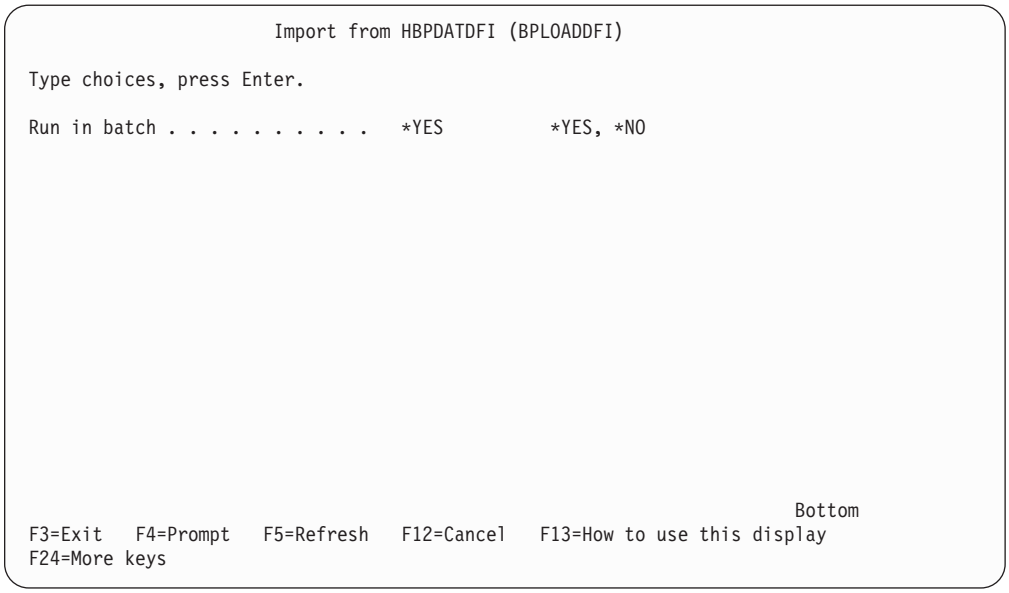
There is a difference in the layout of the interface file HSDATDFI for V3R1M1 and the layout of the interface HSDATDFN for V3R1M2. HSDATDFI is described in “Appendix B. HSDATDFI Interface File (V3R1M1) Layout” on page 183. HSDATDFN is described in “Appendix C. HSDATDFN Interface File (V3R1M2) Layout” on page 187 .

Importing Date-Field Assignment from HSDATDFI

You can import external field assignment that is compatible with a BYPASS2000 V3R1M1 conversion environment from the HSDATDFI file into the active conversion environment. HSDATDFI can be created in several ways. For, example, you can use IBM SEARCH2000 to create the file and export field information to BYPASS2000.

To import database fields that you want to assign from the HSDATDFI file into the corresponding HSDATDFI table in the active environment, do the following:

1. Make sure that you have loaded the user-database information and that the memory-level analysis has completed successfully.
2. Copy the file HSDATDFI that contains the field assignment into the xxxxDB library.
3. Type 11 on the Date Field Assignment menu, to import external date information.



4. Specify whether you want to run this job in batch or interactively.
5. Once the job has completed, check the conversion log to see if BYPASS2000 has encountered any errors while importing the information.
6. Review the imported date field assignment.

Importing Date Field Assignment from HSDATDFN

You can import external field assignment that is compatible with a BYPASS2000 V3R1M2 conversion environment from the HSDATDFN file into the active conversion environment.

Use option 14 of the Field Assignment menu to create HSDATDFN and generate the field assignment from SEARCH2000.

To import database fields that you want to assign from the HSDATDFN file into the corresponding HSDATDFN table in the active environment, do the following:

1. Make sure that you have loaded the user-database information and that the memory-level analysis has completed successfully.
2. Copy the HSDATDFN file that contains the field assignment into the xxxxDB library.
3. Type 12 in the BYPASS2000 Date Field Assignment menu.

```
                      Import from HSDATDFN (BPLOADDFN)

Type choices, press Enter.

Run in batch . . . . . .   *YES          *YES, *NO

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

4. Specify whether you want to run this job in batch or interactively.
5. Once the job has completed, check the conversion log to see whether BYPASS2000 has encountered any errors while importing the information.
6. Review the imported date field assignment.

Creating HSDATDFN from an Existing BYPASS2000 Environment

If you want to reuse date-related information from the active conversion environment in another one, you can export the current field assignment into the HSDATDFN file. You can use this option if you need to create a copy of an existing V3R1M2 environment and do not want to assign date fields manually.

1. Type 13 in the BYPASS2000 Date Field Assignment menu.

```
                    Load HSDATDFN for Export (BPEXPASS1)

Type choices, press Enter.

Environment library  . . . . . .   *LIBL         Name, *LIBL, *CURLIB
Replace or add records . . . . .   *REPLACE      *REPLACE, *ADD
Run in batch . . . . . .   *YES          *YES, *NO

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

2. Specify the name of the new environment, and indicate whether you want to overwrite any existing information in the HSDATDFN file.

3. Execute a CPYF command from the HSDATDFN file of the old environment to the HSDATDFN file of the new environment.
4. Load database information for the new environment (option 2 on the BYPASS2000 for AS/400 Main Menu).
5. Run the memory analysis for the new environment (option 3 on the BYPASS2000 for AS/400 Main Menu.
6. Type 12 in the BYPASS2000 Date Field Assignment menu, to import the date field assignment into the new environment.

The date fields in the new environment are automatically assigned.

Creating HSDATDFN from the SEARCH2000 Repository

You can create the HSDATDFN file directly from information about date fields in database files that was collected by SEARCH2000. This function is important, because SEARCH2000 exports date field assignment in the format of the HSDATDFI file that BYPASS2000 V3R1M1 requires. BYPASS2000 V3R1M2, on the other hand, requires date field information in the format of the HSDATDFN file.

To create the HSDATDFN file directly from information that SEARCH2000 has collected, do the following:

1. Set up a BYPASS2000 environment (option 1 on the BYPASS2000 for AS/400 Main Menu).
2. Load the database information for the new environment (option 2 on the BYPASS2000 for AS/400 Main Menu).
3. Run the memory analysis for the environment (option 3 on the BYPASS2000 for AS/400 Main Menu).
4. Enter the BPIMPDFN command on a command line, or type 14 from the BYPASS2000 Date Field Assignment menu.

```
                  Create Interface File HSDATDFN (BPIMPDFN)

Type choices, press Enter.

Environment library  . . . . . .   *LIBL         Name, *LIBL, *CURLIB
Import Repository Library  . . .                 Name
Replace or add Records . . . . .   *REPLACE      *REPLACE, *ADD
Run in batch . . . . . . . . . .   *YES          *YES, *NO

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

5. For the Import Repository Library parameter, specify which library stores the external field-assignment data.
6. Type 12 in the BYPASS2000 Date Field Assignment menu, to import the field assignment into the active conversion environment.

7. Type 1 in the BYPASS2000 Date Field Assignment menu, to verify the results of the date field assignment in your database files.
8. Type 22 in the BYPASS2000 Date Field Assignment menu, to find out whether any problems may have occurred.

Chapter 13. Propagating Date Fields

Once you have assigned all the date fields, you will need to propagate the available date field information throughout the entire application. The propagation analysis identifies all other date fields in the program code.

Note: Propagation only affects programs; it does not apply to files or COPYs. Propagation does also not extend to fields in database files, which must all be assigned before the propagation analysis can start.

You can propagate date fields either globally or individually.

Choosing between Global and Individual Propagation

The propagation analysis is normally a global process and must run all programs of an application in the same process. Only by analyzing *all* the programs, is it possible to recognize inter-program relationships and propagate year-sensitive fields throughout related programs.

There may be situations in which you can run the propagation analysis for a single program. This may be the case if you make a change in the assignment of some fields in a program and you expect no impact on other programs. In this situation, you may prefer the shorter, individual propagation of the single program to the longer, global propagation of all programs.

For best results you should consider the following:

1. Run the very first propagation analysis globally to allow BYPASS2000 to recognize the basic relationships based on your initial field assignment.

   Note: It is important to have the correct and complete assignment of the year-sensitive fields in your application's database. Fields that have not been assigned correctly, cannot be propagated correctly.
2. If you need to make changes to your conversion environment, or to the assignment of some fields, run an individual propagation analysis for the affected programs.
3. After the propagation analysis of a single program, display the conversion log, and check for requested information to make sure that all newly discovered year-sensitive fields have been propagated correctly.

   You can check the file ANDATOLR, located in the conversion repository, to see if the propagation analysis has discovered new year-sensitive fields that could not be processed, as indicated by the flag FLELAB = 'N'.
4. If you have unresolved issues, you must address them and run the propagation analysis again. You can either run a global propagation analysis, or an individual propagation analysis for the programs involved.
5. You must check the propagation result and repeat the propagation analysis until all open issues have been resolved.

   Note: When you are certain that all open issues have been resolved, run a final global propagation for all programs in you application.

Globally Propagating Date Fields

This approach is normally used for the first propagation analysis of the programs in an environment. To globally propagate date fields for all programs, do the following:

1. Type 5 on the BYPASS2000 for AS/400 Main Menu to access the BYPASS2000 Propagation Analysis display.

```
BP4SPRP               BYPASS2000 Propagation Analysis
                                                          System:   MYAS400
Select one of the following:

     1. Assign date fields

     5. Work with programs
     6. Analyze date field propagation

    22. Display conversion log
    23. Acknowledge requested information

    30. Work with propagation-analysis result
    31. Check propagation trace
    32. Delete propagation-analysis results

    70. Work with submitted jobs
    71. Work with all spooled files
                                                                      Bottom
Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
(C) COPYRIGHT HAL S.p.A. 1994, 1998.
```

2. Type 6 to access the Analyze Date Field Propagation display.

```
                  Analyze Date Field Propagation (BPPRPPGM)

Type choices, press Enter.

From program . . . . . .   *FIRST        Name, *FIRST
To program . . . . . .     *LAST         Name, *ONLY, *LAST
Application code . . . . . .   *ALL       Name, *ALL, *BLANK
Environment library  . . . . . .   *LIBL   Name, *LIBL, *CURLIB
Run in batch . . . . . .   *YES          *YES, *NO

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

3. Enter your parameter selections, and press Enter to start the propagation analysis.

Note: Make sure that you run at least one global propagation analysis of your application to propagate fields between related programs. See “Changing the

Parameter Table" on page 60 for information on the parameters FREE-IPC and SYS-PGM that can force or block inter-program propagation of dates.

At the end of the propagation phase, you should do the following:

1. Display the conversion log (option 22 of the BYPASS2000 Propagation Analysis menu) or type V next to each program in the Work with Program display.
2. Check for requested information (option 23) or type R next to each program in the Work with Program display.
3. Work with propagation result (option 30) or type W next to each program in the Work with Program display.
4. Check propagation trace (option 31) or type T next to each program in the Work with Program display.

Individually Propagating Date Fields

To propagate date fields for individual files, do the following:

1. Type 5 on the BYPASS2000 for AS/400 Main Menu to access the BYPASS2000 Propagation Analysis display.
2. Type 5 to submit propagation for individually selected programs, and to monitor the progress of a propagation job.
   Type P next to each program that you want to propagate or re-propagate.

```
                              Work with Programs
                                                  Position to . . :
 Type choices, press Enter.
   2=Change    4=Delete    5=Display    8=Hold    9=Release
   I=Input source      O=Output source  V=View log             A=Analyze
   D=Delete analysis   P=Propagate      Q=Delete propagation   C=Convert
Opt Pgm Name   Type                      Ana Prp Cnv Gen Log Req Chk
 _  ADU018R    RPG                        *   *   *
 _  ARRAY      RPG                        9   0   0
 _  CALLED     RPG                        9   0   0
 _  CALLER     RPG                        9   0   0
 _  CBLCOPY    CBL                        9   0   0       V
 _  CENDS      RPG                        9   0   0
 _  CENDS1     RPG                        9   0   0
 _  CENDS2     RPG                        9   0   0
 _  CENDS3     RPG                        9   0   0
 _  CPYCON     RPG                        9   0   0       V
 _  CPY1CAL    CBL                        7   0   0       V   Y
 _  CPY1TST    CBL                        9   0   0
 _  DG0100     RPG                        9   0   0
                                                                        More...
 F3=Exit F4=Prompt F5=Refresh F6=Create  F10=Reset flag F11=Toggle F13=Repeat
 F12=Cancel  F21=Command entry  F23=More option      F24=More keys
```

At the end of the propagation, you should do the following:

1. Display the conversion log (option 22 of the BYPASS2000 Propagation Analysis menu) or type V next to the program in the Work with Program display.
2. Check for requested information (option 23) or type R next to the program in the Work with Program display.
3. Work with propagation result (option 30) or type W next to the program in the Work with Program display.
4. Check propagation trace (option 31) or type T next to the program in the Work with Program display.

Performing Propagation for a Subset of Programs

To run the propagation analysis for a subset of the programs in your application, you can proceed in one of the following ways from the Work with Program display:

Global propagation based on propagation flag status '0'

Only programs for which the propagation flag is set to '0' will be analyzed.

1. Set the propagation flags of these programs equal to '0'.
2. Delete the results of the previous propagation analysis for these programs.
3. Exit the Work with Program display.
4. Run a global propagation (option 6, Analyze Program Date Field Propagation, on the BYPASS2000 Propagation-Level Analysis menu)

Global propagation based on propagation flag status '8'

Programs for which the propagation flag is set to '8' will not be analyzed.

1. Set the propagation flags of the programs that you do **not** want to analyze equal to '8'.
2. Delete the results of the previous propagation analysis for the programs that do not have their propagation flag set to '8'.
3. Exit the Work with Program display.
4. Run a global propagation (option 6, Analyze Program Date Field Propagation, on the BYPASS2000 Propagation-Level Analysis menu)

Individual propagation

To propagate individual programs, do the following:

1. Delete the results of the previous propagation analysis for the programs that you want to analyze again.
2. Type `P` next to each program that you want to analyze.
3. Press Enter to start the propagation analysis.

Working with Propagation Result

This function helps you find the fields for which BYPASS2000 could not determine whether or not they are date fields, or what their format should be. These fields are called ″generic fields″.

1. Specify the sources for which you want to view the propagation result, and press Enter.

```
                        Work with propagation result
                             Select parameters


Source name . . . . . . . . . . . . . : *ALL        *ALL, name, generic*


Only sources with

     Propagation incongruences found  : *   (Y/N/*)     (Y=At least one found)
                                                        (N=None found)
     Propagation status forced by user: *   (Y/N/*)     (*=All programs)









F3=Exit  F12=Cancel
```

You can choose to view all sources, or only a subset, such as sources for which BYPASS2000 has found incongruences that need to be resolved. This information is shown in the Select Program display.

```
                                Select Program

                                                     Position to . . :
Type choices, press Enter.
 1=Select   I=Input source  O=Output source
                                                  ---------- Propagation ----------
 Opt Program Name Source File  Library      Incongruence  Changed      Status
  _  ADU018R      QRPGSRC      BP10OLD                                    *
  _  RNP36        QS36PRC      BP10OLD                                    *
  _  SA71094      QRPGLESRC    BP10OLD                                    *
  _  SA71372A     QCBLSRC      BP10OLD                                    *
  _  SA71741A     QRPGSRC      BP10OLD                                    *
  _  SA72721      QRPGSRC      BP10OLD                                    *
  _  SA72721A     QRPGSRC      BP10OLD                                    *
  _  SA72736      QRPGSRC      BP10OLD                                    *
  _  SA72736A     QRPGSRC      BP10OLD                                    *
  _  SORTA        QRPGSRC      BP10OLD                                    *



F3=Exit     F12=Cancel     F17=Top     F18=Bottom
```

2. Type 1 to access the Select Date Field display, and view the program's date fields that BYPASS2000 has found.

```
                          Select Date Field
Program name: TABLE
                                  Position to date field:
 Type choices, press Enter.
  1=Date origin           5=View statement        F=File origin
  D=Date field assigned   I=I/O-area field list   U=Update flag

Opt Src Name   Field Name------------ Type  Len. Date Type-------------- Proc.
 _  TABLE      OUT                   CHAR    40 *   1 YEAR              OK
 _  TABLE      OUT1                  CHAR    40 *   1 YEAR              OK









 F3=Exit  F4=Propag.trace  F5=Refresh   F8=Converted source   F17=Top F18=Bottom
 F12=Cancel  F7=Original source   F19=List assigned date      F20=One/Two Rows
```

Displaying Assigned Dates

To display the assigned dates in a program, do the following:

1. Press F19 in the Select Date Fields display to see which dates you have assigned. The Select Assigned Dates display appears.

```
                           Select Assigned Dates
Program name: TABLE
                            Position to assigned date:
 Type choices, press Enter.
  5=View statement   D=Date field assigned   I=I/O-area field list
  F=File origin

Opt Src Typ/Name Type Date field-------------- Displ  Len. Date Type------------
 _  P TABLE      ZND  TAB                          1     6     2 YEAR,MONTH,DAY
```

```
 F3=Exit  F5=Refresh   F4=Propag.trace  F8=Converted source  F17=Top F18=Bottom
 F12=Cancel    F7=Original source    F19=List propag. date   F20=One/Two Rows
```

   The entry ″generic″ in the Processing column indicates that BYPASS2000 could not properly identify this field as a date during the propagation phase. This could be caused by the field being linked to a year-sensitive field and also to a non-year-sensitive field, or because this field receives more than one date format.

2. To resolve the situation, type `D` to obtain more information about the field. The Assign Date Field - Program I/O-Area List display appears.

```
                  Assign Date Field - Program I/O-Area List
 Pgm name  TABLE
                                Position to . . :
 Type choices, press Enter.
  1=Select   2=Select with lock    *=Annul all assignments and locks
  6=Print assigned dates                      -Assigned-
Opt I/O-Area Name                   Seq       Date  Lock
 _  TAB                             000

                                                                        Bottom
 F3=Exit     F5=Refresh     F12=Cancel
 F17=Top     F18=Bottom     F21=Print assigned date fields of listed sources
```

3. Type 1 to select (assign) the field, or type 2 to select and lock the field.

Displaying Date Origin

Type 1 next to a field in the Select Date Field display. The Display Date Origin display appears.

```
                           Display Date Origin
 Pgm name : TABLE                          Source type/name: PGM TABLE
 I/O area : OUT
 Fld. name: OUT                            CHAR Len.    40  Disp.     1
 Date type:   1 YEAR                *SSTR  Year length/pos./off.  2     1     1
 Type choices, press Enter.             Position to date:
  1=Full data     5=View statement
  F=File origin   I=I/O area field list   T=Date tree                      P St
Opt D T Src Name   Field type/name              Len.  Data type            F Nu
 _  Y P TABLE      ZND  TAB                        6     2 YEAR,MONTH,DAY Y 01









 F3=Exit     F4=Propagation trace     F7=Original source     F8=Converted source
 F12=Cancel  F17=Top                  F18=Bottom             F20=One/Two Rows
```

This display shows the fields that propagate different information to field APP. In this sample display, dates for which BYPASS2000 has found incongruences are highlighted. Dates that have been clearly identified as being related to user-assigned dates are flagged by a 'Y' in the Date column.

From this display, you can access detailed information about each field, as shown in the following displays:

Display Full Date Information

Type 1 next to a field in the Display Date Origin display to view full date information.

```
                        Display Full Date Information
Press Enter to continue.
          Original date type (FROM)      ---> Propagated date type (TO)
Date type:   2 YEAR,MONTH,DAY                  1 YEAR
Origin...:     *PRELOAD-ASSIGNED-FIELD
Year length:    2 At Pos.     1               Year length 2   at pos.     1

          Original date field (FROM)     ---> Propagated date field (TO)
Src name : *SAME                              TABLE
I/O area : TAB                                OUT

Field....: TAB                                OUT
   Type..: ZND                                CHAR   *SUBSTR
   Displ.:     1                                  1
   Length:     6                                 40

   OCCURS:    20
Propagation status . . : Y Propagation OK
Propagation step number:     1
                         Destination I/O-area block
Start in original I/O area:     1        Start in propagated I/O area:     1
Length of I/O-area block  :    40

F3=Exit    F12=Cancel
```

Display File Origin

Type F next to a field in the Display Date Origin display to view information about the field's file origin and, if necessary, to assign the field as a date field. This would resolve any incongruences that may have been found during propagation.

```
                  ...........................................................
 Pgm name : TABLE  :                   File Origin                         :
 I/O area : OUT    : Source name TABLE        Internal field name TAB       :
 Fld. name: OUT    : Type choices, press Enter.                              :
 Date type:   1 YE :  D=Date field assigned                                  : 1
 Type choices, pre :                    Position to . . :                   :
  1=Full data      :                                                         :
  F=File origin    :                        External   External   Physical  : t
Opt D T Src Name   : Opt Filename           Field      Record     Filename   : u
 F  Y P TABLE      :                                                         : 1
                   :                                                         :
                   :                                                         :
                   :                                                         :
                   :                                                         :
                   :                                                         :
                   :                                                         :
                   :                                                         :
                   :                                                         :
                   :                                                         :
                   :  F3=Exit   F12=Cancel   F17=Top   F18=Bottom            :
 F3=Exit     F4=Pr :  No data to list                                       : e
 F12=Cancel  F17=T :                                                       :
                   :...........................................................:
```

Display I/O-Area Field List

Type I next to a field in the Display Date Origin display to access information about the I/O area that corresponds to this date field.

```
                          Display I/O-Area Field List
 Pgm name: TABLE
 I/O area: TAB                                              Seq.   0

                                                         Position to Displ:
 Type choices, press Enter.
 1=Select   5=Display all field attributes
                                                                  G S OCC
Opt Field Name                                 Displ Len.  Type Int Dec F F Nbr.
 _  01  TAB                                        1     6 ZND    6   0        20









 F3=Exit          F5=Refresh          F12=Cancel
 F17=Top          F18=Bottom
```

Type 5 next to a field in the Display I/O Area – Field List display to access detailed information about the field attributes.

```
                          Display All Field Attributes
 Press Enter to continue.
 Source type : PGM   Source name : TABLE       Stmt Number :       1  Sect.: E

 I/O area attributes
  Name . . . . : TAB                            Seq.    0            Nmbr.     2

 Field attributes
  Name . . . . : TAB                            Disp.     1          Nmbr.     2
  Level  . . . : 01             Group field                Type ZND
  Length . . . :     6          Integer/dec.    6 /   0    Sign     Just.

 OCCURS and REDEFINES field attributes
  OCCURS . . . :    20          Total OCCURS     20
  OCCURS level :     1          OCCURS seq.       1         Shift     6
  REDEFINES  . :                REDEFINES nbr     0         Clause

 Date field and new attributes
  Date type  . :   1    YEAR  YEAR
  Year length  : 2
  New displ. . :     1          New length        8

 F3=Exit  F12=Cancel
```

Working with Propagation Tree

Type B next to a field in the Display Date Origin display to see the entire propagation tree between the selected field and field APP.

```
                          Work with Propagation Tree
 Program name: TABLE
 Type choices, press Enter.                        Expand to level :  00
 "+"=Expand  "-"=Collapse 5=Display field information   6=Display date position

Opt Field  Field name                                                        P
    lev                                                                      F
 _  01 - TAB
 _  02   . OUT
 _  02   . OUT1
 _  02   . TAB
 _  02   . TAB
 _  02   . TAB
 _  02   . TAB
 _  02   . TAB
 _  02   . TAB
 _  02   . TAB
 _  02   . TAB
 _  02   . TAB
 _  02   . TAB                                                              +

 F3=Exit   F5=Refresh   F7=Redraw all levels  F8=Redraw second level  F12=Cancel
 F17=Top   F18=Bottom   F10=View locked       F11=View generic
```

Highlighted fields indicate incongruences that must be resolved.

Forcing the Propagation of a Generic Field

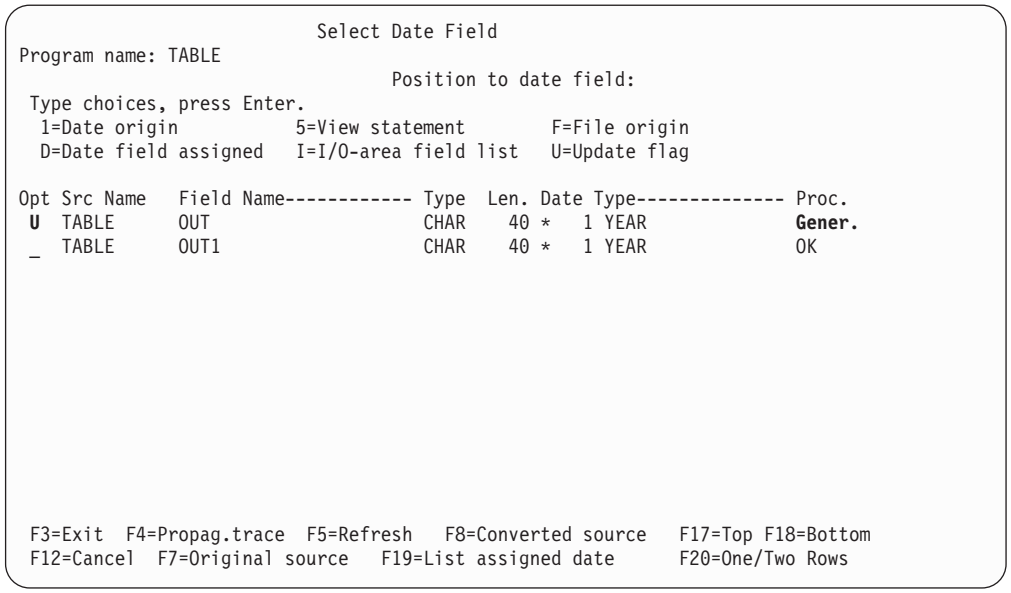
If BYPASS2000 discovers generic areas in certain programs during the propagation analysis, proceed with the following:

1. Resolve the generic areas through the assignment of additional fields, in the programs.
2. Run the propagation analysis again.

This approach does not solve the problem if the generic field is connected to a database field that contains a date (for example assigned as date type 003), and to a database field that does not contain a date.

If there are no other generic areas, do the following:

1. Type `U` next to the generic field in the Select Date Field display.



This forces the field to be the same date type 003 as the year-sensitive database field that it is connected with. This automatically resets the propagation flag from 9 to 7.

2. Run the propagation analysis again.

   BYPASS2000 treats the forced field as if it had been discovered during the propagation analysis.

You can use this approach when you do not want to resolve generic areas through additional field assignment. A new propagation analysis from flag status 0 would take much longer than from flag status 7.

Note: You should use this approach only if you know the application very well and are absolutely sure about the possible connections of the generic field.

Checking the Propagation Trace

Wait until the propagation analysis has completed (indicated by the flag '9' in the Propagation Status column in the Work with Program display) before you verify the propagation trace. Use the Check Propagation Trace function (option 31 on the Propagation Analysis menu) to view information about how BYPASS2000 has propagated date field information throughout your application, and to find all incongruent relationships between fields for each program.

You can also check the propagation trace by simply typing T next to an entry in the Work with Program display.

```
                        Display Propagation Trace

Type choices, press Enter.                       Position to . . :
  1=Select

 Opt Program Name Source File  Library      Ana.   Prop.  Conv. Incongruence

  _  ADU018R      QRPGSRC      BP10OLD        9      9      *       Y
  _  RNP36        QS36PRC      BP10OLD        9      9      *       N
  _  SA71094      QRPGLESRC    BP10OLD        9      9      *       N
  _  SA71372A     QCBLSRC      BP10OLD        9      9      *       N
  _  SA71741A     QRPGSRC      BP10OLD        9      9      *       N
  _  SA72721      QRPGSRC      BP10OLD        9      9      *       N
  _  SA72721A     QRPGSRC      BP10OLD        9      9      *       N
  _  SA72736      QRPGSRC      BP10OLD        9      9      *       N
  _  SA72736A     QRPGSRC      BP10OLD        9      9      *       N
  _  SORTA        QRPGSRC      BP10OLD        9      9      *       N




F3=Exit      F12=Cancel        F17=Top      F18=Bottom
```

1. Type 1 next to a program name in this display to access a list of all the date fields in the program.

```
                    Source name : TABLE                          Page 01
    I/O Area              Field                  Date Field


--->TAB                   TAB                    TAB
                                                 Displ:     1 Pos:     1  Y
--->TAB                   TAB                    TAB
                                                 Displ:     7 Pos:     1  Y
--->TAB                   TAB                    TAB
                                                 Displ:    13 Pos:     1  Y
--->TAB                   TAB                    TAB
                                                 Displ:    19 Pos:     1  Y
--->TAB                   TAB                    TAB
                                                 Displ:    25 Pos:     1  Y
--->TAB                   TAB                    TAB
                                                 Displ:    31 Pos:     1  Y
--->TAB                   TAB                    TAB
                                                 Displ:    37 Pos:     1  Y
--->TAB                   TAB                    TAB
                                                 Displ:    43 Pos:     1  Y
                                                                    More...
F3=Exit      F4=Expand input/Display entry    F5=Initial list
F12=Cancel   F10=Statement list               F15=Display only incongruences
```

Note: An arrow (- - ->) to the left of a field indicates propagation leading into this field.

2. To see the propagation trace for a field, put your cursor on the field name and press F4.

```
                    Source name : TABLE                              Page 01
:.....................................................................:
:                                                                     :
:                      -------- FROM FIELDS --------                 :
:                                                                     :
:                                                                     :
:                                                                     :
:                                                                     :
: FROM Field   : *PRELOAD-ASSIGNED-FIELD                              :
:                                                                     :
:                                                                     :
:                                                                     :
:                     --------- TO FIELDS ---------                  :
:                                                                     :
:                                                                     :
:                                                  Fld. len: 00006    :
:                                                  *...+....1....+....2.... :
: TO   Field   : TAB                               YY----             :
:                                                                     :
:                                                                     :
:                                                                     :
:      F3=Exit      F12=Cancel                                        :
:                                                                     :
:.....................................................................:
```

3. To display only dates for which incongruences exist, press F15 in the Source Name display.

```
                    Source name : CINS                               Page 01
   I/O Area              Field                  Date Field


  I DRHDTE               DRHDTE                 DRHDTE
                                                Displ:     1 Pos:     5--->
  I DATE2                DATE2                  DATE2
                                                Displ:     1 Pos:     1--->









                                                                      Bottom
F3=Exit      F4=Expand input/Display entry    F5=Initial list
F12=Cancel   F10=Statement list               F15=Display only incongruences
```

Note: An arrow (- - ->) to the right of a field indicates propagation originating from this field.

Deleting Propagation Results

After having solved any propagation problems, you must run the propagation analysis again. Before you do this, you must first delete the result of the previous propagation.

1. Type 32 on the BYPASS2000 Propagation Analysis menu to delete the propagation analysis.

```
                  Delete Date Field Propagation (BPDLPPGM)

Type choices, press Enter.

From program . . . . . .   *FIRST        Name, *FIRST
To program . . . . . .     *LAST         Name, *ONLY, *LAST
Application code . . . . . .  *ALL       Name, *ALL, *BLANK
Environment library  . . . . . .  *LIBL   Name, *LIBL, *CURLIB
Reorganize File  . . . . . .  *NO         *YES, *NO
Run in batch . . . . . .   *YES          *YES, *NO

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

2. Select the parameters of your choice and press Enter.

Propagating Arithmetic Operations That Involve Years

If BYPASS2000 is uncertain about an arithmetic operation involving year data that has not been assigned as a date, it issues a message such as ″Unknown propagation effect″. If you do not take any action, BYPASS2000 highlights the operation after the conversion, without propagating the result.

You must verify the operation logic in the source, and either lock or assign and lock the variable that BYPASS2000 was not able to identify. After any date assignment, you must delete the result of the previous propagation analysis, and run the propagation again.

Note: The same problem as with arithmetic operations involving years also occurs with STRING operations in COBOL/400, and must be handled in the same manner.

Chapter 14. Repeating a Processing Phase

Depending on the circumstances, you may have to repeat one or more phases of the BYPASS2000 processing. If you need to repeat an analysis phase (memory or propagation), do the following:

1. Delete the previous analysis results for this phase.

   For example, if you need to repeat the analysis of DDS source, you must delete both the old DDS analysis result and the analysis results for the programs that use this DDS source.
2. Run the required analysis again.

   For example, if you need to repeat the analysis of DDS source, must run both the DDS analysis and the analysis of the programs that use this DDS source.

If you need to repeat the conversion phase for all DDS, COPYs, or programs, simply reset the corresponding conversion flag(s) to 0, without any deletions. There is no need to reset the conversion flag to 0 if you want to repeat the conversion phase for a single DDS, COPY, or program.

Using Delete Functions

Delete functions are available for each phase. These functions have the following effect:

- Delete the results of a previous analysis
- Set the corresponding status flags to 0
- Delete the corresponding unchecked messages in the conversion log
- Delete the corresponding unchecked requests for user information
- Delete both the propagation trace and any related generic areas (when the propagation analysis is deleted).

Note: You must not manually set the status flags for memory and propagation analysis to 0, but use the appropriate delete functions.

The following table shows the impact of each deletion.

Table 6. Deletion – Impact after Memory-Level Analysis

Type	Results of Deletion
Memory-level program analysis	• Remove program analysis results • Remove any propagation results • Set ANA flag to 0 • Set PROP flag to 0 • Set CONV flag to 0
Memory-level COPY and program analysis	• Remove program analysis results • Remove COPY analysis results • Remove any propagation results • Set ANA flag to 0 for program and COPY • Set PROP flag to 0 for program and COPY • Set CONV flag to 0 for program and COPY

Table 6. Deletion – Impact after Memory-Level Analysis (continued)

Type	Results of Deletion
Propagation-level analysis	• Remove results of program date field propagation • Remove results of CLP date field propagation (when global) • Set the PROP flag to 0 • Set the CONV flag to 0

Chapter 15. Changing the Conversion Environment

The following sections list the steps that may be required when making changes to the conversion environment.

Note: If you have created a customized conversion environment, your naming convention may be different.

Adding COPY Members

1. Load the COPY members in the QCPYSRC and/or QLBLSRC file of the old source library *xxxx*OLD.
2. Add the members to the COPY list of sources, as described in “Analyzing COPYs” on page 80.

Adding SQL Table Members

1. Load the members in the QSQLSRC file of the old source library*xxxx*OLD, and the objects in the old object library*xxxx*DAT.
2. Perform the Load AS/400 Database Information function again, as described in “Chapter 8. Loading User-Database Information” on page 77.
3. Add the members to the SQL list of sources, as described in “Analyzing SQL Source” on page 83.

Adding Physical Files

1. Add the physical file to the old object library xxxxDAT.

   (If you have created a customized conversion environment, your naming convention may be different.)
2. Type 2 on the BYPASS2000 for AS/400 Main Menu to load the database information again.
3. Perform the memory analysis for the new physical file.
4. Perform the field assignment for the new physical file.
5. Convert the new physical file.

Adding a Program

1. Add the program to one of the following files in the old source library:
   - xxxxOLD/QCLPSRC
   - xxxxOLD/QCBLSRC
   - xxxxOLD/QRPGSRC
2. Press F24 in the Work with Program display to include the new member.
3. Perform the memory analysis for the new program.
4. If necessary, perform the field assignment for the new program.
5. Run the program date field propagation for the new program.

   This step is useful to find out whether the program is related to any other programs that have so far not been identified.

Note: If the program is related to other programs through CALL statements, the propagation analysis must be run for the subset of all related programs.

6. Convert the new program and all related programs.

Adding Program Source

1. Load the members in the QRPGSRC, QCLPSRC, or QCBLSRC file of the old source library *xxxx*OLD.
2. Add the members to the program list of sources, as described in “Analyzing Programs” on page 84.

Modifying COPY Members

1. Remove the following:
   a. Any date-field assignments.
   b. Relationship with the corresponding files.
2. Delete the memory analysis results for the following:
   a. All programs using the COPY.

      Note: This will also remove any propagation analysis results.
   b. The COPY itself.
3. Perform the memory analysis for the following:
   a. The changed COPY.
   b. All programs related to the changed COPY.
4. Relate the changed COPY to the corresponding files.
5. Assign date fields in the changed COPY.
6. Run the program date field propagation for COPY and programs.

   This step is useful to find out if the COPY is related to areas of any other programs that have not been identified.
7. Convert the following:
   a. The COPY.
   b. All related programs.

Modifying a Physical File

1. Remove any date-field assignments.
2. Delete memory analysis results for all programs, COPYs and CLP using the physical file. This will also remove any propagation analysis results.
3. Load the database information again (option 2 on the BYPASS2000 for AS/400 Main Menu).
4. Perform the memory analysis for the changed physical file, the programs, and the CLP.
5. Assign date fields in the changed physical file.
6. Run the program date field propagation for COPY, CLP, and programs.
7. Convert the physical file, the COPY, the programs, and the CLP.

Modifying Program Source

1. Replace each modified member in the corresponding file (QRPGSRC, QCLPSRC, or QCBLSRC) in the *xxxx*OLD library.

Note: For COBOL programs, delete the first normalization line that was inserted during the initial analysis of the program.

```
******* BYPASS2000 NORMALIZATION
```

2. From the BYPASS2000 Field Assignment menu type `6` to print the assigned field list.
3. Select the parameters of your choice in the Assigned Date Field List display and press Enter.
4. Print the field assignment for the changed program.

   Note: If you modify a structure by inserting or deleting a field and this structure requires date-field assignment after the modification, the new source would find the assignment in the incorrect position. BYPASS2000 will delete the incorrect assignment during the new analysis and issue a message for each deleted assignment.
5. Delete the analysis of the modified programs, using option 'D' in the Work with Program display.
6. Analyze the modified programs as described in "Analyzing Programs" on page 84 . (If you use the Analyze Program option, BYPASS2000 will analyze only the modified programs indicated by the status flag set to '0'.)
7. Check the conversion log (option 22).
8. Check requested information (option 23).

   In case any field assignment has been deleted, type `2` in the BYPASS2000 Field Assignment menu to correct the assignment.

Modifying a Source Program

1. Remove any date-field assignment for the working-storage fields of the program.
2. Modify the source.
3. Delete program memory analysis results. This will also remove any propagation analysis results.
4. Perform the memory analysis for the modified program.
5. If necessary, assign date fields in the modified program.
6. Run the program date field propagation for the modified program.
7. Convert the modified program and all related programs.

Modifying the Assignment of a Date Field

1. Change assignment of fields in physical file, RPG, CBL, CLP, or COPY.
2. Delete propagation results for programs, COPY, and CLP.
3. Perform the program date field propagation.
4. Convert the program again.

Removing Program Source

If you need to remove any program source from the conversion environment, do the following:

1. In the corresponding Work with ... display, type the option 'D' to cancel the analysis of such members.
2. Type `4` to remove the entry from the list of sources.

Note: If you remove programs from the source list, BYPASS2000 automatically deletes all date-field assignments for these programs.

3. Remove the corresponding source members from the appropriate file (QRPGSRC or QCBLSRC) in the *xxxx*OLD library.

Removing Files, COPYs, or SQL Table Definitions

If you need to remove any file, COPY, or SQL table members from the conversion environment, do the following:

1. Cancel the analysis of such members in the corresponding Work with ... display.
2. Type 4 to remove the entry from the list of sources.

   Note: If you remove members from the source list, BYPASS2000 automatically deletes all date-field assignment for these members.
3. Remove the file objects from the old object library *xxxx*DAT.
4. Remove the corresponding source members from the appropriate file (QDDSSRC, QCPYSRC, or QSQLSRC) in the *xxxx*OLD library.

Chapter 16. Converting Source Code

Type 6 on the BYPASS2000 for AS/400 Main Menu to access the BYPASS2000 Application Conversion menu.

```
BP4SCNV              BYPASS2000 Application Conversion
                                                              System:   TORAS002
Select one of the following:

     1. Work with database information
     2. Convert DDS sources
     3. Work with COPY sources
     4. Convert COPY sources
     5. Work with programs
     6. Convert program sources
     7. Convert DDS,COPY, and program sources
     8. Convert logical files
     9. Work with new dictionary names
    10. Work with files

    12. Create migration programs
    13. Create migration-dispatcher program

                                                                        More...
Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
(C) COPYRIGHT HAL S.p.A. 1994, 1998.
```

From the conversion menu, you can select options to convert the various parts of your application.

- For individual conversions, use the Work with ... options.

```
                              Work with Database Information
                                                 Position to . . :
 Type choices, press Enter.
   2=Change    4=Delete    5=Display    8=Hold    9=Release
   I=Input source   O=Output source   V=View log  E=Reset cnv. flag K=Reset chk
   A=Analyze  D=Delete analysis  C=Convert   F=Override files    J=Dynamic call
Opt DDS  Name                            Ana     Cnv Gen Log Req Chk
 _  ADSDEMO                               9       9       V
 _  ADU006P                               9       9
 _  ADU007P                               9       9
 _  ADU009P                               9       9
 _  ADU011P                               9       9
 _  ADU020P                               9       9
 _  ADU049P                               9       9
 _  CENDSP                                9       9       V
 _  CENDSP1                               9       9
 _  FPO                                   9       9
 _  GETB01LA                              9       9
 _  HHAVLX                                9       9
 _  I00221                                9       9
                                                                        More...
 F3=Exit F4=Prompt F5=Refresh F6=Create  F10=Reset flag F11=Toggle
 F12=Cancel F15=Include new member       F21=Command entry      F24=More keys
```

- For global conversions, use the Convert ... options.

```
                          Convert DDS (BPCNVDDS)

Type choices, press Enter.

From DDS . . . . . .   *FIRST        Name, *FIRST
To DDS . . . . . .     *LAST         Name, *ONLY, *LAST
Application code . . . . . .   *ALL   Name, *ALL, *BLANK
Environment library  . . . . . .   *LIBL   Name, *LIBL, *CURLIB
Run in batch . . . . . .   *YES       *YES, *NO

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Specify the parameters of your choice and press Enter.

- To convert all different parts of your application with a single request, use the Convert DDS, COPY, and Program option.
  Specify the parameters of your choice and press Enter.

```
                    Convert COPYs and Programs (BPCNVSRC)

Type choices, press Enter.

Application code . . . . . .   *ALL   Name, *ALL, *BLANK
Environment library  . . . . . .   *LIBL   Name, *LIBL, *CURLIB
Run in batch . . . . . .   *YES       *YES, *NO

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Note: If you do not use option 2 (Convert DDS) or option 7 (Convert DDS, copy and program), remember to convert your dictionaries after all other files.

- Monitor the conversion progress, or look at the output source (options 1, 3, and 5).
- Check the conversion log for messages (option 22).
- To create migration programs, test-migration programs, and date-integrity modules, use the appropriate Create... options in the BYPASS2000 Application Conversion menu.

  The various utility programs and their creation are described in “Chapter 17. Testing the Converted Application Using Utility Programs” on page 147.

- Review the converted source code to see the markers BYPASS2000 has inserted as comments. See “Appendix D. Markers Added to Converted Sources” on page 189 for a list of these markers and their meaning.

Chapter 17. Testing the Converted Application Using Utility Programs

From the BYPASS2000 Application Conversion menu, you can create utility programs for testing a converted application. You can create these programs either for individual files, using option 10 (Work with files), or globally, using the Create ... menu options). In addition, you can create dispatcher programs for all types of utility programs.

Creating Utility Programs

BYPASS2000 can create the following utility programs for the all files:

- Migration programs
- Test-migration programs
- DIM programs.

Creating Utility Programs for Individual Files

To create utility programs for individual files, do the following:

1. Type 1 to access the Work with File – Select Parameters display.

```
                              Work with File
                            Select parameters




Filename  . . . . . : *ALL                *ALL, name, generic*

Application code  : *ALL                  *ALL, name, generic*


Display only files with:

  Program type  . : *                     * = All programs
                                          M = Migration programs
    Not started   (0) :  Y                T = Test migration programs
    In progress (1-8) :  Y                D = DIM programs
    Completed     (9) :  Y


F3=Exit   F12=Cancel
```

2. Specify the parameters of your choice and press Enter.

```
                                Work with File
                                          Position to file type .
 Type choices, press Enter.               Position to filename  .
   2=Change       4=Delete     5=Display      8=Hold      9=Release
   M=Migrate      T=Test       D=DIM          S=Output program
   V=View log
Opt File Type   Filename           Migr Test  DIM       History    Appl. Code
 _  PF          ADSDEMO              0    0    0
 _  PF          I00221               0    0    0
 _  PF          LOGF                 0    0    0
 _  PF          REUSE                0    0    0
 _  PF          SA69226              0    0    0
 _  PF          SA69298P             0    0    0
 _  PF          SA69982              0    0    0
 _  PF          SA69988              0    0    0
 _  PF          SA69997              0    0    0
 _  PF          SA70077              9    0    0        M
 _  PF          SA70302              0    0    0
 _  PF          SA70715              0    0    0
 _  PF          SA71710              0    0    0                           More...

 F3=Exit  F4=Opt prompt  F5=Refresh  F6=Create  F10=Reset flag  F13=Repeat
 F11=Work with pgm. name     F12=Cancel    F21=Command entry    F24=More keys
```

3. Type one of the following options next to each file for which you want to create a utility program and press Enter:
   - `M` to create a migration program
   - `T` to create a test-migration program
   - `D` to create a date-integrity module

   Note: You must repeat this step if you want to create more than one type of utility program for a file.

   After the successful creation of the utility programs, the Migration, Test, and DIM status flags are set to 9.
4. To view one of the utility programs that was created, type `S` next to a file name and press Enter.
5. Select which utility program you want to work with from the pop-up menu.

   The following display shows part of a test-migration program.

```
Columns . . . :    1  71          Browse                    BP01NEW/QSBRSRC
SEU==>                                                          HT000002
FMT *  ..... *. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
       ********** Beginning of data **************************************
0001.00       * BYPASS2000 - "TEST" Migration program for file STOCK
0002.00       *
0003.00       * To be used only for TEST purposes.
0004.00       * Do not use for production.
0005.00      FFILEIN  IP  F      48            DISK
0006.00      FFILEOUT O   F      49            DISK
0007.00      IFILEIN  AA
0008.00      I                                      1  25 ND0001
0009.00      I                                   P  26  290DI0001
0010.00      I                                      30  48 ND0002
0011.00      C                    MOVE '00713'   HB2@FA
0012.00      C                    MOVE DI0001    HB2@F
0013.00      C                    MOVE '009P3'   HB2@TA
0014.00      C                    EXSR HB@ADD
0015.00      C                    MOVE HB2@T     DO0001  90
0016.00      C                    EXCPTFM0001

F3=Exit   F5=Refresh   F9=Retrieve   F10=Cursor   F11=Toggle   F12=Cancel
F16=Repeat find        F24=More keys
                                     (C) COPYRIGHT IBM CORP. 1981, 1995.
```

Creating Migration Programs for All Files

To create migration programs for all files in your application, do the following:

1. Type 12 on the BYPASS2000 Application Conversion menu.

```
                    Create Migration Programs (BPMGRPGM)

Type choices, press Enter.

Language . . . . . .   RPG          RPG, COBOL
File Type  . . . . . .   PF           PF
From file  . . . . . .   *FIRST
To file  . . . . . .   *LAST
Environment library  . . . . . .   *LIBL         Name, *LIBL, *CURLIB
Run in batch . . . . . .   *YES         *YES, *NO

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

2. Specify the parameters of your choice and press Enter.

Creating Test Migration Programs for All Files

To create test migration programs for all files in your application, do the following:

1. Type 15 on the BYPASS2000 Application Conversion menu.

```
                  Create Test Migration Programs (BPTSTPGM)

Type choices, press Enter.

Language . . . . . .   RPG          RPG, COBOL
File Type  . . . . . .   PF           PF
From file  . . . . . .   *FIRST
To file  . . . . . .   *LAST
Environment library  . . . . . .   *LIBL         Name, *LIBL, *CURLIB
Run in batch . . . . . .   *YES         *YES, *NO

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

2. Specify the parameters of your choice and press Enter.

Creating Date Integrity Module (DIM) Programs for All Files

To create DIM programs for all files in your application, do the following:

1. Type 18 on the BYPASS2000 Application Conversion menu.

```
                      Create DIM Programs (BPDIMPGM)

Type choices, press Enter.

Language . . . . . .   RPG          RPG, COBOL
File Type  . . . . . .   PF           PF
From file  . . . . . .   *FIRST
To file  . . . . . .   *LAST
Environment library  . . . . . .   *LIBL         Name, *LIBL, *CURLIB
Run in batch . . . . . .   *YES         *YES, *NO

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

2. Specify the parameters of your choice and press Enter.

Creating Dispatcher Programs

For each type of utility program, BYPASS2000 can create a corresponding dispatcher program.

Migration Dispatcher Program

The migration dispatcher program is a CLP program that creates the OVRDBF on FILEIN and FILEOUT, and performs a CALL to each migration program. BYPASS2000 places this program called HMIGRBCH in the file QSBRSRC in the new source library specified during the creation of the conversion environment.

After the migration dispatcher program has performed calls to all migration programs, it calls the program HBP2CPYFIL for each file that does not contain dates that need to be migrated. HBP2CPYFIL performs a CPYF command from the original file to the same file in the new object library. You can customize the HBP2CPYFIL program or disable calls to HBP2CPYFIL by simply removing the /* and */ characters from the line `/* GOTO CMDLBL(ENDTAG) */` in the migration dispatcher program.

To create the migration dispatcher program, do the following:

1. Type `13` on the BYPASS2000 Application Conversion menu.

```
                    Create Migration Dispatcher (BPMGDPGM)

Type choices, press Enter.

Environment library  . . . . . .   *LIBL         Name, *LIBL, *CURLIB
Run in batch . . . . . . . . . .   *YES          *YES, *NO

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

2. Specify the parameters of your choice and press Enter.

See “Migrating Database Files” on page 153 for an illustration of the steps required to migrate database files.

Test-Migration Dispatcher Program

The test-migration dispatcher program is a CLP program that creates the OVRDBF on FILEIN and FILEOUT and performs a call to each test migration program. BYPASS2000 places this program called HTESTBCH in the file QSBRSRC in the new source library specified during the creation of the conversion environment.

After the dispatcher program has performed calls to all test migration programs, it calls the program HBP2CPYFIL for each file that does not contain dates that need to be test-migrated. HBP2CPYFIL performs a CPYF command from the original file to the same file in the new object library. You can customize the HBP2CPYFIL

program or disable calls to HBP2CPYFIL by simply removing the /* and */ characters from the line `/* GOTO CMDLBL(ENDTAG) */` in the test migration dispatcher program.

To create test-migration dispatcher programs for all files in your application, do the following:

1. Type `16` on the BYPASS2000 Application Conversion menu.

```
                  Create Test Migr. Dispatcher (BPTSDPGM)

Type choices, press Enter.

Environment library  . . . . . .   *LIBL         Name, *LIBL, *CURLIB
Run in batch . . . . . . . . . .   *YES          *YES, *NO

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

2. Specify the parameters of your choice and press Enter.

See “Testing the Migration of Database Files” on page 154 for an illustration of the steps required to test the migration of database files after the conversion of their DDS.

Customizing the HBP2CPYFIL Program: The source code for the program HBP2CPYFIL is available in file QCLPSRC of library QBP2000. Originally the program performs a CPYF command with the MBRPROT(*REPLACE) and FMTOPT(*NOCHK) parameters. You can change these parameters or change the CPYF with the CRTDUPOBJ command. After the changes, you must compile HBP2CPYFIL again and run the migration or test migration dispatcher program again.

DIM Dispatcher Program

To create a DIM dispatcher program, do the following:

1. Type `19` on the BYPASS2000 Application Conversion menu.

```
                      Create DIM Dispatcher (BPDIDPGM)

Type choices, press Enter.

Environment library  . . . . . .   *LIBL         Name, *LIBL, *CURLIB
Run in batch . . . . . . . . . .   *YES          *YES, *NO

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

2. Specify the parameters of your choice and press Enter.

Examples

The following examples illustrate how to:

- Create migration programs and a migration dispatcher program to migrate database files.
- Create test migration programs and a test-migration dispatcher program to test the migration of database files.

Migrating Database Files

The following example shows the steps required to migrate two database files TESTPF1 and TESTPF2, after the conversion of their DDS.

1. Compile the converted DDS TESTPF1 and TESTPF2, and put the resulting objects in the new object library (specified during the creation of the conversion environment).
2. Type 2 on the BYPASS2000 Database Conversion Menu, to create the programs HM000001 and HM000002 in file QSBRSRC, in the new source library (specified during the creation of the conversion environment).
3. Type 3 on the BYPASS2000 Database Conversion menu, to create the program HMIGRBCH in file QSBRSRC, in the new source library.

   BYPASS2000 creates the following program:

```
/*           BYPASS200 RUNTIME MIGRATION PROGRAM          */
/*                                                        */
PGM
OVRDBF  FILE(FILEIN) TOFILE(XXXXDAT/TESTPF1)
OVRDBF  FILE(FILEOUT) TOFILE(XXXXOBJ/TESTPF1)
CALL    PGM(HM000001)
DLTOVR  FILE(FILEIN)
DLTOVR  FILE(FILEOUT)
OVRDBF  FILE(FILEIN) TOFILE(XXXXDAT/TESTPF2)
CALL    PGM(HM000002)
DLTOVR  FILE(FILEIN)
DLTOVR  FILE(FILEOUT)
        .
        .
/* GOTO   CMDLBL(ENDTAG)    */

 CALL       PGM(HBP2CPYFIL)  PARM(FILNAME +
                 XXXXDAT   XXXXOBJ)
        .
        .
ENDTAG:      ENDPGM
```

where xxxxDAT is the old object library and xxxxOBJ is the new object library.

4. Compile the migration programs HM000001 and HM000002, and the CLP program HMIGRBCH (with QBP2000 in your library list).

   If you have generated migration programs with BYPASS2000 V3R1M1, perform the following steps:

   a. Add your new object library to your library list.
   b. For RPG applications, copy member HBP2CVTR from file QRPGSRC in library QBP2000 to file QRPGSRC in the new source library.

      For COBOL applications, copy member HBP2CVTPM and HBP2CVTWM from file QLBLSRC in library QBP2000 to file QLBLSRC in the new source library.
   c. For RPG applications, place the compiled COPYs HBP2CVT and HBP2DTS from library QBP2000 in your new object library.

      For COBOL applications, place the compiled COPYs HBP2CVTP and HBP2DTS from library QBP2000 in your new object library.

      Note: If you change the default setting for the SHIFT parameter, compile the source HBP2DTS from file QCLPSRC in library QBP2000, and place it in the new object library.
   d. Compile the migration programs and the migration dispatcher program from QSBRSRC in the new source library and place them in your new object library.
   e. Add the new object library to your library list.
5. Issue the command `CALL xxxxOBJ/HMIGRBCH` to migrate your database.

Testing the Migration of Database Files

The following example shows the steps required to test the migration of two database files TESTPF1 and TESTPF2, after the conversion of their DDS.

1. Compile the converted DDS TESTPF1 and TESTPF2, and put the resulting objects in the new object library (specified during the creation of the conversion environment).

   Tip: Skip this step if you have already compiled the converted DDS as part of “Migrating Database Files” on page 153.

2. Type 4 on the BYPASS2000 Database Conversion menu to create the programs HT000001 and HT000002 in file QSBRSRC in the new source library (specified during the creation of the conversion environment).
3. Type 5 on the BYPASS2000 Database Conversion menu to create the program HTESTBCH in file QSBRSRC in the new source library .

   BYPASS2000 creates the following program:

```
 /*            BYPASS200 RUNTIME TEST MIGRATION PROGRAM      */
 /*                                                          */
 PGM
 OVRDBF  FILE(FILEIN) TOFILE(XXXXDAT/TESTPF1)
 OVRDBF  FILE(FILEOUT) TOFILE(XXXXOBJ/TESTPF1)
 CALL    PGM(HT000001)
 DLTOVR  FILE(FILEIN)
 DLTOVR  FILE(FILEOUT)
 OVRDBF  FILE(FILEIN) TOFILE(XXXXDAT/TESTPF2)
 CALL    PGM(HT000002)
 DLTOVR  FILE(FILEIN)
 DLTOVR  FILE(FILEOUT)
         .
         .
/* GOTO   CMDLBL(ENDTAG)    */

 CALL       PGM(HBP2CPYFIL)                  PARM(FILNAME +
              XXXXDAT XXXXOBJ)
         .
         .
ENDTAG:      ENDPGM
```

   where xxxxDAT is the old object library and xxxxOBJ is the new object library.
4. Compile the test migration programs HT000001 and HT000002 and the CLP program HTESTBCH:
   a. Add your new object library to your library list.
   b. For RPG applications and programs created with BYPASS2000 V3R1M1, copy member HBP2CVTR from file QRPGSRC in library QBP2000 to file QRPGSRC in the new source library.

      Tip: Skip this step if you have already copied HBP2CVTR as part of the “Migrating Database Files” on page 153.

      For COBOL applications, copy members HBP2CVTPM and HBP2CVTWM from file QLBLSRC in library QBP2000 to file QLBLSRC in the new source library.
   c. If you have not customized the environment COPYs, you must make a change to the CLP program HBP2DTS in file QBP2000/QCLPSRC in order to shift the dates in the database.

```
00001.00 /*BYPASS2000 - YEAR SHIFT FOR TEST
00002.00 PGM  &YSHIFT
00003.00 DCL  &SHIFT*CHAR 2
00004.00 CHGVAR &SHIFT VALUE('00')
00005.00 ENDPGM
```

      Substitute the number of years by which you want to shift the database for ’00’ in VALUE(’00’), and recompile HBP2DTS.

      Note: If you create more than one version of the HBP2DTS program, you must ensure that you are using the proper version for the test-data generation versus the system-date shifting, by managing your job’s library list.

d. For RPG applications, place the compiled programs HBP2CVT and HBP2DTS from library QBP2000 in your new object library.

   For COBOL applications, place the compiled programs HBP2CVTPM, HBP2CVTWM, and HBP2DTS from library QBP2000 in your new object library.

e. Compile the test migration programs and the test-migration dispatcher program located in QSBRSRC in the new source library, and place them in your new object library.

5. Add the new object library to your library list.
6. Issue the command `CALL xxxxOBJ/HTESTBCH` to test-migrate your database.

Working with DIM Programs

The following example shows the steps required to compile and use DIM programs.

1. For RPG applications only, copy members HBP2DIM1 and HBP2DIM2 from the file QRPGSRC in library QBP2000 to the file QRPGSRC in your new library xxxxY2K.
2. For both RPG and COBOL applications, compile the DIM programs and the DIM dispatcher program from the QSBRSRC file in the library where you put your output source files for DIM modules.
3. Add your object library to your library list.
4. Attach the trigger to the physical files:
   a. For each physical file, type the command ADDPFTRG (Add Physical File Trigger) and specify the following parameters:

      ```
      ADDPFTRG FILE(LIBXXX/FILE1) TRGTIME(*BEFORE)
         TRGEVENT(*INSERT) PGM(QBP2000/HBP2TRG)
      ```

   b. Again type the command ADDPFTRG, and specify the following parameters:

      ```
      ADDPFTRG FILE(LIBXXX/FILE1) TRGTIME(*BEFORE)
         TRGEVENT(*UPDATE) PGM(QBP2000/HBP2TRG)
      ```

   Note: You can add the program HBP2TRG to your user library to avoid the use of the library QBP2000. In this case, specify the name of your user library instead of library QBP2000 as a parameter on the ADDPFTRG command.

   After attaching the trigger to the physical file (for example, when you perform a DFU operation), the trigger will check dates that you add to, or change in the physical file. If the trigger program finds the date format to be incorrect, a message will be issued.

Decimal Data Error Issued from Trigger DIM Program

If you use the EXCPT instruction to add or partially update a record from an RPG program instruction, you may receive a decimal data error from a trigger DIM program. The error appears when not all seeded date fields are used in the output statement for the EXCPT instruction.

The following example illustrates this case:

```
DDS for file FILIN
A          R RECIN
A            CAM01          1A
A            AN01           2S 0 --> seeded as type 001 Year
A            CAM02         10A
A            AN02           2S 0 --> Seeded as type 001 Year

Source program RPGIN1
FFILIN   O   F      15           DISK
C                     MOVE 'C'       CAM01   1
C                     MOVE 'CCCCCCCC'CAM02  10
C                     EXCPTADD
C                     SETON                     LR
OFILIN   EADD             ADD
O                         CAM01      1
O                         CAM02     13

Source program RPGIN
FFILIN   UF  F      15           DISK
IFILIN   NS  01
I                                       1   1 CAM01
I                                       4  13 CAM02
C                     READ FILIN                  99
C           *IN99     IFEQ '0'
C                     MOVE 'A'       CAM01
C                     MOVE 'BBBBBBBB'CAM02
C                     EXCPTUPDATE
C                     ENDIF
C                     SETON                     LR
OFILIN   E                UPDATE
```

To avoid this error, specify the following parameter when you compile the DIM program:

```
Ignore decimal data error  . . .   *YES
```

Chapter 18. Compiling the Converted Application Source

After the conversion of your application source, you must compile the application before you can run it. The following source needs to be included in the compilation.

Source File: QCBLSRC	
Source Member	Description
HBP2TRG	The HBP2TRG program object source (Date-Integrity Control Trigger).
HBP2CVTP	This Copybook must be inserted in converted COBOL38 source. It adds and removes century information while the converted application is running, and shifts the system date. It calls the HBP2CVT program.
HBP2CVTW	This Copybook must be inserted in converted COBOL38 source. It is the working storage area for the HBP2CVTP, HBP2CVTP34, and HBP2CVTP23 Copybooks.
HBP2CVTP23	This Copybook must be inserted in converted COBOL38 source. It adds and removes century-flag information while the converted application is running, and shifts the system date. It calls the HBP2CV program.
HBP2CVTP34	This Copybook must be inserted in converted COBOL38 source, to handle relationships between years with a full century and years with a century flag, while the application is running. It calls program HBP2CV.
HBP2CVTFP	This Copybook contains the COBOL38 USA programs for year-reversal. It is included with the converted source to handle the year-reversal in date fields on instructions such as COMPUTE DAT1 = 10000.01 * DAT2.
HBP2CVTFW	This Copybook contains the working storage area for the HBP2CVTFP Copybook. It is included in the converted COBOL38 source.
HBP2000W	This Copybook contains a generic working storage area that is used for IF statement conversion. It is included in the converted COBOL 38 source.
HBP2DIMW	This Copybook is inserted in the COBOL DIM programs that were created with V3R1M1.

Source File: QS36SRC	
Source Member	**Description**
HBP2CVTP	This Copybook must be inserted in converted COBOL36 source. It adds and removes century information while the converted application is running, and shifts the system date. It calls program HBP2CV.
HBP2CVTW	This Copybook must be inserted in converted COBOL36 source. It is the working storage area for the HBP2CVTP, HBP2CVTP34 and HBP2CVTP23 Copybooks.
HBP2CVTP23	This Copybook must be inserted in converted COBOL36 source. It adds and removes century-flag information while the converted application is running, and shifts the system date. It calls program HBP2CV.
HBP2CVTP34	This Copybook must be inserted in converted COBOL36 source, to handle the relationships between years with full century information, and years with a century flag while the converted application is running. It calls program HBP2CV.

Source File: QS36SRC	
Source Member	**Description**
HBP2CVTFP	This Copybook contains the COBOL36 USA year-reversal programs. It is included with the converted source to handle the year-reversal in date fields on instructions such as COMPUTE DAT1 = 10000.01 * DAT2
HBP2CVTFW	This Copybook contains the working storage area for the HBP2CVTFP Copybook. It is included in the converted COBOL36 source.
HBP2000W	This Copybook contains a generic working storage area that is used for IF statement conversion. It is included in the converted COBOL 36 source.
HBP2CVTF	This RPG36 /COPY contains the USA year-reversal programs. It is included with the converted source to handle the year-reversal in date fields on instructions such as 10000.01 MULT DAT1 DAT2 or 100.0001 MULT DAT1 DAT2.
HBP2CVT6	This RPG36 /COPY contains the USA year-reversal programs. It is included with the converted source to handle the year-reversal in date fields on instructions such as 100.01 MULT DAT1 DAT2 or 100.01 MULT DAT1 DAT2.
HBP2CVTR	This Copybook must be inserted in RPG36 source converted with BYPASS2000 V3R1M1. It is used to add and remove century information while the converted application is running, and shifts the system date. It handles the most common supported date types, and calls HBP2CVT for the conversion of the less common supported date types.
HBPNSE	This Copybook contains the array definitions (E specs) for the HBPNSC Copybook. It is included in the converted RPG36 source.
HBPNSI	This Copybook contains the field definitions (I specs) for the HBPNSC Copybook. It is included in the converted RPG36 source.
HBPNSC	This Copybook must be inserted in converted RPG36 source. It adds and removes century information while the converted application is running, and shifts the system date. It handles non-standard data types.
HBPSIM	This Copybook must be inserted in converted RPG36 source, to add century information while the converted application is running. It handles the most common data types (data type 001, 002 and 003).
HBP23E	This Copybook contains the array definitions (E specs) for the HBP23C Copybook. It is included in the converted RPG36 source.
HBP23I	This Copybook contains the field definitions (I specs) for the HBP23C Copybook. It is included in the converted RPG36 source.
HBP23C	This Copybook must be inserted in converted RPG36 source. It adds and removes century-flag information while the converted application is running, and shifts the system date.
HBP24E	This Copybook contains the array definitions (E specs) for the HBP23C Copybook. It is included in the converted RPG36 source.
HBP24I	This Copybook contains the field definitions (I specs) for the HBP23C Copybook. It is included in the converted RPG36 source.
HBP24C	This Copybook must be inserted in converted RPG36 source. It adds and removes century information while the converted application is running, and shifts the system date. It handles standard data types.
HBP34E	This Copybook contains the array definitions (E specs) for the HBP23C Copybook. It is included in the converted RPG36 source.

Source File: QS36SRC	
Source Member	**Description**
HBP34I	This Copybook contains the field definitions (I specs) for the HBP23C Copybook. It is included in the converted RPG36 source.
HBP34C	This Copybook must be inserted in converted RPG36 source, to handle relationships between years with full century information, and years with a century flag while the converted application is running.

Source File: QLBLSRC	
Source Member	**Description**
HBP2CVTP	This Copybook must be inserted in converted COBOL source. It adds and removes century information while the converted application is running, and shifts the system date.
HBP2CVTW	This Copybook must be inserted in converted COBOL source. It is the working storage area for the HBP2CVTP, HBP2CVTP34 and HBP2CVTP23 Copybooks
HBP2CVTP23	This Copybook must be inserted in converted COBOL source. It adds and removes century flag information while the converted application is running, and shifts the system date
HBP2CVTP34	This Copybook must be inserted in converted COBOL source, to handle relationships between years with full century information and years with a century flag while the converted application is running.
HBP2CVTFP	This Copybook contains the COBOL USA year-reversal programs. It is included with the converted source to handle the year-reversal in date fields on instructions such as COMPUTE DAT1 = 10000.01 * DAT2
HBP2CVTFW	This Copybook contains the working storage area for the HBP2CVTFP Copybook. It is included in the converted COBOL source
HBP2000W	This Copybook contains a generic working storage area used for IF statement conversion. It is included in the converted COBOL source
HBP2DIMW	This Copybook is inserted in the COBOL DIM programs that were created with V3R1M1.
HBP2CVTPM	This Copybook must be inserted in COBOL migration and test migration programs, to add a century flag, to add full century information, and shifts dates.
HBP2CVTWM	This Copybook is included in COBOL migration and test migration programs. It is the working storage area for the HBP2CVTPM Copybook.

Source File: QS38SRC	
Source Member	**Description**
HBP2CVTF	This RPG36 /COPY contains the USA year-reversal programs. This Copybook is included with the converted source to handle the year-reversal in date fields on instructions such as 10000.01 MULT DAT1 DAT2 or 100.0001 MULT DAT1 DAT2.
HBP2CVT6	This RPG38 /COPY contains the USA year-reversal programs. This Copybook is included with the converted source to handle the year-reversal in date fields on instructions such as 100.01 MULT DAT1 DAT2 or 100.01 MULT DAT1 DAT2.

Source File: QS38SRC	
Source Member	**Description**
HBP2CVTR	This Copybook must be inserted in RPG38 source converted with V3R1M1 release in order to add and remove century information while the converted application is running, and shifts the system date. It handles the most common supported date types, and calls HBP2CVT for the conversion of the less common supported date types.
HBPNSE	This Copybook contains the array definitions (E specs) for the HBPNSC Copybook. It is included in the converted RPG38 source.
HBPNSI	This Copybook contains the field definitions (I specs) for the HBPNSC Copybook. It is included in the converted RPG38 source.
HBPNSC	This Copybook must be inserted in converted RPG38 source. It adds and removes century information while the converted application is running, and shifts the system date. It handles the non-standard data types.
HBPSIM	This Copybook must be inserted in converted RPG38 source, to add century information while the converted application is running. It handles the most common data types (data type 001, 002 and 003).
HBP23E	This Copybook contains the array definitions (E specs) for the HBP23C Copybook. It is included in the converted RPG38 source.
HBP23I	This Copybook contains the field definitions (I specs) for the HBP23C Copybook. It is included in the converted RPG38 source.
HBP23C	This Copybook must be inserted in converted RPG38 source. It adds and removes century-flag information while the converted application is running, and shifts the system date.
HBP24E	This Copybook contains the array definitions (E specs) for the HBP23C Copybook. It is included in the converted RPG38 source.
HBP24I	This Copybook contains the field definitions (I specs) for the HBP23C Copybook. It is included in the converted RPG38 source.
HBP24C	This Copybook must be inserted in converted RPG38 source. It adds and removes century information while the converted application is running, and shifts the system date. It handles standard data types.
HBP34E	This Copybook contains the array definitions (E specs) for the HBP23C Copybook. It is included in the converted RPG38 source.
HBP34I	This Copybook contains the field definitions (I specs) for the HBP23C Copybook. It is included in the converted RPG38 source.
HBP34C	This Copybook must be inserted in converted RPG38 source, to handle relationships between years with full century information and years with a century flag while the converted application is running.

Source File: QRPGLESRC	
Source Member	**Description**
HBP2CVTF	This RPGILE /COPY contains the USA year-reversal programs. It is included with the converted source to handle the year-reversal in date fields on instructions such as 10000.01 MULT DAT1 DAT2 or 100.0001 MULT DAT1 DAT2.
HBP2CVT6	This RPGILE /COPY contains the USA year-reversal programs. It is included with the converted source to handle the year-reversal in date fields on instructions such as 100.01 MULT DAT1 DAT2 or 100.01 MULT DAT1 DAT2.

Source File: QRPGLESRC	
Source Member	**Description**
HBP2CVTR	This Copybook must be inserted in RPGILE source converted with V3R1M1 release in order to add and remove century information while the converted application is running, and shifts the system date. It handles the most common supported date types, and calls HBP2CVT for the conversion of the less common supported date types.
HBPNSD	This Copybook contains the field definitions (D specs) for the HBPNSC Copybook. It is included in the converted RPGILE source.
HBPNSC	This Copybook must be inserted in converted RPGILE source. It adds and removes century information while the converted application is running, and shifts the system date. It handles the non-standard data types.
HBPSIM	This Copybook must be inserted in converted RPGILE source, to add century information while the converted application is running. It handles the most common date types (date type 001, 002 and 003).
HBP23D	This Copybook contains the field definitions (D specs) for the HBP23C Copybook. It is included in the converted RPGILE source.
HBP23C	This Copybook must be inserted in converted RPGILE source. It adds and removes century-flag information while the converted application is running, and shifts the system date.
HBP24D	This Copybook contains the field definitions (D specs) for the HBP23C Copybook. It is included in the converted RPGILE source.
HBP24C	This Copybook must be inserted in converted RPGILE source. It adds and removes century information while the converted application is running, and shifts the system date. It handles standard data types.
HBP34D	This Copybook contains the field definitions (D specs) for the HBP23C Copybook. It is included in the converted RPGILE source.
HBP34C	This Copybook must be inserted in converted RPG36 source, to handle relationships between years with full century information and years with a century flag, while the converted application is running.

Source File: QRPGSRC	
Source Member	**Description**
HBP2CVTF	This RPG /COPY contains the USA year-reversal programs. It is included in the converted source to handle the year-reversal in date fields on instructions such as 10000.01 MULT DAT1 DAT2 or 100.0001 MULT DAT1 DAT2.
HBP2CVT6	This RPG /COPY contains the USA year-reversal programs. It is included in the converted source to handle the year-reversal in date fields on instructions such as 100.01 MULT DAT1 DAT2 or 100.01 MULT DAT1 DAT2.
HBP2CVTR	This Copybook must be inserted in RPG source that were converted with BYPASS2000 V3R1M1. It adds and removes century information while the converted application is running, and shifts the system date. It handles the most common supported date types, and calls HBP2CVT for the conversion of the less common supported date types.
HBP2CVTRN	This Copybook must be inserted in migration and test migrations programs that were created with BYPASS2000 V3R1M1. It calls HBP2CVTN to add or remove the century while the converted application is running. It also performs the year shift on file fields.

Source File: QRPGSRC	
Source Member	**Description**
HBP2CVT	This is the source for the runtime program HBP2CVT that adds and removes the century while the converted application is running. It also performs the year shift on the system date.
HBP2CVTN	This is the source for the runtime program HBP2CVTN that adds and removes the century while the migration and test migration programs are running. It also performs the year shift.
HBP2DIM1	This RPG /COPY contains DIM area definitions. It is used during the compilation of DIM programs that were created with BYPASS2000 V3R1M1.
HBP2DIM2	This RPG /COPY contains DIM date-checking programs. It is used during the compilation of DIM programs that were created with BYPASS2000 V3R1M1.
HBP2DIM1N	This RPG /COPY contains DIM area definitions. It is used during the compilation of DIM programs that were created with BYPASS2000 V3R1M1, after applying PTF SA70077.
HBP2DIM2N	This RPG /COPY contains DIM date-checking programs. It is used during the compilation of DIM programs that were created with BYPASS2000 V3R1M1 after applying PTF SA70077.
HBP2DIMC	This RPG /COPY contains DIM date-checking programs. It is used during the compilation of DIM programs.
HBP2RPGMTE	This Copybook contains the array definitions (E specs) for the HBP2RPGMTC Copybook. It is included in the migration and test migration programs.
HBP2RPGMTI	This Copybook contains the field definitions (I specs) for the HBP2RPGMTC Copybook. It is included in the migration and test migration programs.
HBP2RPGMTC	This Copybook must be inserted in migration and test migration programs in order to add a century flag or a full century and shifts dates.
HBPNSE	This Copybook contains the array definitions (E specs) for the HBPNSC Copybook. It is included in the converted RPG source.
HBPNSI	This Copybook contains the field definitions (I specs) for the HBPNSC Copybook. It is included in the converted RPG source.
HBPNSC	This Copybook must be inserted in converted RPG source. It adds and removes century information while the converted application is running, and shifts the system date. It handles non-standard data types.
HBPSIM	This Copybook must be inserted in converted RPG source, to add century information while the converted application is running. It handles the most common data types (data type 001, 002 and 003).
HBP23E	This Copybook contains the array definitions (E specs) for the HBP23C Copybook. It is included in the converted RPG source.
HBP23I	This Copybook contains the field definitions (I specs) for the HBP23C Copybook. It is included in the converted RPG source.
HBP23C	This Copybook must be inserted in converted RPG source. It adds and removes century-flag information while the converted application is running, and shifts the system date.
HBP24E	This Copybook contains the array definitions (E specs) for the HBP23C Copybook. It is included in the converted RPG source.

Source File: QRPGSRC	
Source Member	**Description**
HBP24I	This Copybook contains the field definitions (I specs) for the HBP23C Copybook. It is included in the converted RPG source.
HBP24C	This Copybook must be inserted in converted RPG source. It adds and removes century information while the converted application is running, and shifts the system date. It handles standard data types.
HBP34E	This Copybook contains the array definitions (E specs) for the HBP23C Copybook. It is included in the converted RPG source.
HBP34I	This Copybook contains the field definitions (I specs) for the HBP23C Copybook. It is included in the converted RPG source.
HBP34C	This Copybook must be inserted in converted RPG source, to handle relationships between years with full century information and years with a century flag, while the converted application is running.

Chapter 19. Packaging the Converted Application

The converted application must have access to the following objects to run:

HBP2TRG *PGM
COBOL program to trigger date-integrity control. (You only need this object if you have created DIM programs, as described in “Working with DIM Programs” on page 156.)

HBP2DTS *PGM
CL program to shift year information for testing purposes. (You only need this object in order to test how the application behaves at the turn of the century, as described in “Migrating Database Files” on page 153.)

Applications that have been converted with BYPASS2000 V3R1M1 must have access to the following object to run:

HBP2CVT *PGM
RPG program to add, remove, and shift century information.

Migration and test migration programs that you have created using BYPASS2000 V3R1M1 with PTF SA70077 must have access to the following object to run:

HBP2CVTN *PGM
RPG program to add, remove, and shift century information.

For future maintenance of your converted application, you also need access to the source members listed in “Chapter 18. Compiling the Converted Application Source” on page 159.

Removing Date-Shifting Logic

Before putting your converted application into production, you must remove any date-shifting logic that you may have added by specifying ’Y’ for the parameter ″Insert SHIFT instruction on system date″, during the creation of your conversion environment (see “Choosing the Right Conversion Parameters” on page 8).

You may choose to do this by manually modifying the application code, or by changing the parameter value to ’N’ and running the final conversion of your application again.

After removing the date-shifting logic, you must recompile your application, as described in “Chapter 18. Compiling the Converted Application Source” on page 159.

Part 3. Reference Information

Chapter 20. Assigning Date Fields in Program-Described Files

BYPASS2000 can work with files that have program descriptions instead of DDS or SQL description. There are multiple ways to accomplish this. The most common way is to create a Copybook member with source code describing the record format.

If you have a file that contains year-sensitive fields in different positions, you can identify these fields using their relative positions (if you know them). It is easier to work with the layout to the file and identify the date-sensitive fields in the layout.

Note: BYPASS2000 requires that the layout must be defined in a Copybook, and not in the program. It is not necessary to put a Copy directive in any of your program source. You only need a separate source member containing the record description.

Creating a Copybook

A file can have different types of records, such as header records and detail records. BYPASS2000 views each type of record as a record format. The definition of the fields in a record format is called a layout. The same record can have different layouts in Copybooks, INCLUDEs, or inside the program.

If a separate Copybook does not exist for each file in which date fields are to be assigned, you must create one. The layout must be copied from the program in which the layout is defined. You can create a temporary Copybook, if a real one does not exist.

Note: It is not necessary to remove the copied layout from the program, because BYPASS2000 will update the existing program layout as well as the newly created Copybook.

Because multiple layouts can exist for a record format, as record layouts within the program or as Copybooks, each record format must be defined in a Copybook. You must provide enough information to uniquely identify the displacement of each variable containing year information for each record format.

Note: BYPASS2000 works on the displacement of an assigned field, not on its name. Therefore, date data is always in the same position, even if the field layout is different.

Once you have created the COPYs proceed as described below. It is important to perform the steps in the order in which they are listed.

1. Add the new COPYs to the file QCPYSRC in library *xxxx*OLD (or in the equivalent old source library for type CPY, if you have created a customized conversion environment).
2. Perform the memory analysis for the new COPYs.
3. Create a relationship between the I/O areas in the COPYs and the corresponding files (option 3, Assign I/O Area to Related File, on the BYPASS2000 Field Assignment menu).
4. Remove the old relationships that exist for the file.
5. Assign the date-sensitive fields in the new COPYs (option 1, Assign date field for I/O Area Related to File, on the BYPASS2000 Field Assignment menu).

During the subsequent propagation phase, this field assignment will be propagated to all related programs.

Once the conversion of your application has been successful, you can delete the Copybooks that were created for correctly assigning and propagating dated fields.

Handling Files with Multiple Record Formats

If you have a file that contains different record formats, and every record format is a REDEFINES of a previous one (with the date field in different positions in each record format), you must define every record format.

Each multiple-format file must have its own record type. The application database is the starting point of the propagation, and an assignment error caused by different positions of date fields in each record format will be propagated everywhere. Furthermore, if the same record type is assigned to different record formats, you could lose some date fields during the migration of files.

Note: If none of the record formats contains year-sensitive fields, you can ignore this step.

If you have assigned a file to more than one logical area (multiple-format file), and one of these layouts contains date fields, you cannot be sure that the information will be propagated to the fields that are located in the same position in the other layouts. As well, when generating migration programs for these files, it is necessary to know in which layout the century information needs to be added or left untouched. BYPASS2000 provides a record-type assignment function to assign different record-type values for each record format of a file, to determine in which layout the century must be assigned.

Note: The date field propagation will not start if there are multiple-format files without an assigned record type.

If an RPG program contains a definition of a multiple-format file, you must do the following:

1. Create a COPY (or one COPY for each format) to define the multiple-format file in *xxxx*OLD/QCPYSRC.
2. Perform the memory analysis for each COPY.
3. Relate the I/O area to the related file (from the Assign I/O Area to Related File display).
4. Delete the original relationship.
5. Assign the record type for each format (from the Assign Record Type to Related I/O Area display.
6. Assign the date fields in the program (from the Assign Date Field display).
7. Assign the date fields in each COPY (from the Assign Date Field for I/O Area Related to File display).

If you have implemented the incorrect COPY for the multiple-format file and want to delete it, you must first delete the field assignment of the COPY and dissociate the file from the I/O area. Then you must delete the analysis of the COPY, and finally the COPY itself.

If the record type of each format is not sequential and not completely defined, BYPASS2000 cannot build the migration program.

Chapter 21. Converting Dictionaries

When BYPASS2000 converts an application that makes use of dictionaries (reference files), it lets you decide how to convert the dictionary. You do so by specifying the value of the conversion parameter ″Create new dictionary (’Y’) or convert existing dictionary (’N’). See “Choosing the Right Conversion Parameters” on page 8 for a description of how to use this conversion parameter.

The following list shows what results you can expect for the different parameter values.

Create new dictionary

If you choose to create a new dictionary, BYPASS2000 creates a dictionary with the default name HBPDIZ. BYPASS2000 adds one new field to this dictionary for each field that has been assigned in the original dictionaries. This field has a new name and is referenced to the original field. Consequently, the new dictionary HBPDIZ contains fields for all assigned date fields from all dictionaries in the conversion environment.

BYPASS2000 puts the new dictionary HBPDIZ in the new source library *xxxx*Y2K (or the equivalent file, if you have created a customized conversion environment). You cannot see the file from within the BYPASS2000 environment. The only indication that the new dictionary is present is that the files which contain fields referenced to the dictionary now show the reference to the new dictionary.

Convert existing dictionary - modify the expanded fields

If you choose to convert the existing dictionary and specify that you want to modify the expanded fields, BYPASS2000 modifies the fields definitions directly in the dictionary and does not make any changes in the files that refer to the dictionary.

Convert existing dictionary - add after existing field

If you choose to convert the existing dictionary and specify that you want to add new fields after the existing fields, BYPASS2000 inserts one new field for each field that has been assigned in the original dictionaries after the definition of the field in the dictionary itself. This field has a new name and is referenced to the original field.

Convert existing dictionary - add at end

If you choose to convert the existing dictionary and specify that you want to add new fields at the end, BYPASS2000 appends one new field for each field that has been assigned in the original dictionaries at the end of the dictionary itself. This field has a new name and is referenced to the original field.

Example

In the following example, the conversion environment contains the dictionary DCTALIAS. DCTALIAS contains four date fields:

- DTINVOICE is assigned DMY (year length is 2, expansion type is 4)
- DTPROM is assigned DMY (year length is 2, expansion type is 3)
- FYY is seeded Year (year length is 2, expansion type is 4)
- TYY is seeded Year (year length is 2, expansion type is 3)

Before conversion, the dictionary DCTALIAS appears as:

```
A          R DALIASR                    TEXT('DATA DEFINITION.')
A            DTINVOICE      8P00       ALIAS(INVOICE_DATE)
A                                       COLHDG('ENTRY' 'DATE')
A                                       TEXT('ENTRY DATE')
A            DTPROM         8P00       ALIAS(PROMISED_SHIP_DATE)
A                                       COLHDG('PROMISED' 'SHIP DAT
A                                       TEXT('PROMISED SHIP DATE')
A            DTSHIP         8P00       ALIAS(SHIP_DATE)
A                                       COLHDG('SHIP' 'DATE')
A                                       TEXT('SHIP DATE')
A            FYY            2S00       COLHDG('FROM YEAR')
A                                       TEXT('FROM YEAR')
A            TYY            2S00       COLHDG('TO YEAR')
A                                       TEXT('TO YEAR')
```

Fields in the dictionary DCTALIAS are used by the file F1ALIAS:

```
A                                       REF(DCTALIAS)
A          R F1ALIASR                   TEXT('DATE')
A
A            DTINVOICE R
A            DTPROM    R
A            FYY       R
A            TYY       R
```

The following date fields have been assigned in the dictionary, and spread to all files that reference the dictionary:

- DTINVOICE (Date type=003, Year length=2 Expansion type=4)
- DTPROM (Date type=003, Year length=2 Expansion type=3)
- FYY (Date type=001, Year length=2 Expansion type=4)
- TYY (Date type=001, Year length= 2 Expansion type=3)

Results for option 'Create a new dictionary':

The dictionary DCTALIAS remains unchanged after the conversion:

```
DCTALIAS
      A          R DALIASR                    TEXT('DATA DEFINITION.')
      A            DTINVOICE      8P00       ALIAS(INVOICE_DATE)
      A                                       COLHDG('ENTRY' 'DATE')
      A                                       TEXT('ENTRY DATE')
      A            DTPROM         8P00       ALIAS(PROMISED_SHIP_DATE)
      A                                       COLHDG('PROMISED' 'SHIP DATA
      A                                       TEXT('PROMISED SHIP DATE')
      A            DTSHIP         8P00       ALIAS(SHIP_DATE)
      A                                       COLHDG('SHIP' 'DATE')
      A                                       TEXT('SHIP DATE')
      A            FYY            2S00       COLHDG('FROM YEAR')
      A                                       TEXT('FROM YEAR')
      A            TYY            2S00       COLHDG('TO YEAR')
      A                                       TEXT('TO YEAR')
```

In the new source library, BYPASS2000 creates a new dictionary HBPDIZ that contains one field for each date-related field from the dictionary DCTALIAS. The fields in HBPDIZ have been renamed, and are mapped to the corresponding ones in the original dictionary DCTALIAS:

```
    R HBPDIZ
     A     BP20000001R   +2                REFFLD(DTINVOICE DCTALIAS)
     A     DTPROM2000R   +1                REFFLD(DTPROM DCTALIAS)
     A     FYY2000   R   +2                REFFLD(FYY DCTALIAS)
     A     TYY2000   R   +1                REFFLD(TYY DCTALIAS)
```

The date fields in the new dictionary have been renamed as follows:

- DTINVOICE is called B20000001
- DTPROM is called DTPROM2000
- FYY is called FYY2000
- TYY is called TYY2000

Note: DTINVOICE used a different naming convention than the other fields. If the filename followed by the suffix 2000 exceeds the maximum length allowed for filenames, BYPASS2000 changes the original filename to BP2000, followed by a 4-digit incremental suffix, starting at 0001.

After the conversion, the converted file F1ALIAS contains references that map the date fields to the corresponding fields in the new dictionary HBPDIZ:

```
F1ALIAS
     A                                  REF(DCTALIAS)
     A          R F1ALIASR              TEXT('DATE')
     A
     A
B2MODA            DTINVOICE R          REFFLD(BP20000001 HBPDIZ)
B2MODA            DTPROM    R          REFFLD(DTPROM2000 HBPDIZ)
B2MODA            FYY       R          REFFLD(FYY2000 HBPDIZ)
B2MODA            TYY       R          REFFLD(TYY2000 HBPDIZ)
```

Results for option 'Convert existing dictionary - modify expanded fields:

The dictionary DCTALIAS is changed after the conversion:

```
DCTALIAS
      A          R       DALIASR           TEXT('DATA DEFINITION.')
B2MODA                   DTINVOICE 10P00   ALIAS(INVOICE_DATE)
     A                                     COLHDG('ENTRY_DATE')
     A                                     TEXT('ENTRY DATE')
B2MODA                   DTPROM     9P00   ALIAS(PROMISED_SHIP_DATE)
     A                                     COLHDG('PROMISEDSHIPDATA')
     A                                     TEXT('PROMISED SHIP DATE')
     A                   DTSHIP     8P00   ALIAS(SHIP_DATE)
     A                                     COLHDG('SHIP_DATE')
     A                                     TEXT('SHIP DATE')
B2MODA                   FYY        4S00   COLHDG('FROM YEAR')
     A                                     TEXT('FROM YEAR')
B2MODA                   TYY        3S00   COLHDG('TO YEAR')
     A                                     TEXT('TO YEAR')
```

After the conversion, the converted file F1ALIAS remains unchanged.

Results for option 'Convert existing dictionary - add after existing field:

The dictionary DCTALIAS is changed after the conversion:

```
DCTALIAS
     A          R      DALIASR              TEXT('DATA DEFINITION.')
     A                 DTINVOICE     8P00   ALIAS(INVOICE_DATE)
     A                                      COLHDG('ENTRY_DATE')
     A                                      TEXT('ENTRY DATE')
B2NEWA                   BP20000001  R   +2  REFFLD(DTINVOICE *SRC)
     A                   DTPROM      8P00   ALIAS(PROMISED_SHIP_DATE)
     A                                      COLHDG('PROMISEDSHIPDATA')
     A                                      TEXT('PROMISED SHIP DATE')
B2NEWA                   DTPROM2000 R   +1  REFFLD(DTPROM   *SRC)
     A                   DTSHIP     8P00    ALIAS(SHIP_DATE)
     A                                      COLHDG('SHIP_DATE')
     A                                      TEXT('SHIP DATE')
     A                   FYY        2S00    COLHDG('FROM YEAR')
     A                                      TEXT('FROM YEAR')
B2NEWA                   FYY2000    R   +2  REFFLD(FYY *SRC)
     A                   TYY        2S00    COLHDG('TO YEAR')
               A                            TEXT('TO YEAR')
B2NEWA                   TYY2000    R   +1  REFFLD(TYY *SRC)
```

The date fields in the converted dictionary have been renamed as follows:

- DTINVOICE is called B20000001
- DTPROM is called DTPROM2000
- FYY is called FYY2000
- TYY is called TYY2000

After the conversion, the converted file F1ALIAS contains references that map the date fields to the corresponding fields in the converted dictionary:

```
    F1ALIAS
         A                                REF(DCTALIAS)
         A     R F1ALIASR       TEXT('DATE')
         A
         A
    B2MODA       DTINVOICE R REFFLD(BP20000001)
    B2MODA       DTPROM    R REFFLD(DTPROM2000)
    B2MODA       FYY       R REFFLD(FYY2000)
    B2MODA       TYY       R REFFLD(TYY2000)
```

Results for option 'Convert existing dictionary - add after existing field:

The dictionary DCTALIAS is changed after the conversion:

```
DCTALIAS
     A    R  DALIASR           TEXT('DATA DEFINITION.')
     A       DTINVOICE   8P00  ALIAS(INVOICE_DATE)
     A                         COLHDG('ENTRY_DATE')
     A                         TEXT('ENTRY DATE')
     A       DTPROM      8P00  ALIAS(PROMISED_SHIP_DATE)
     A                         COLHDG('PROMISEDSHIPDATA')
     A                         TEXT('PROMISED SHIP DATE')
     A       DTSHIP      8P00  ALIAS(SHIP_DATE)
     A                         COLHDG('SHIP_DATE')
     A                         TEXT('SHIP DATE')
     A       FYY         2S00  COLHDG('FROM YEAR')
     A                         TEXT('FROM YEAR')
     A       TYY         2S00  COLHDG('TO YEAR')
     A                         TEXT('TO YEAR')
     A*
B2NEWA       BP20000001 R  +2  REFFLD(DTINVOICE *SRC)
B2NEWA       DTPROM2000 R  +1  REFFLD(DTPROM *SRC)
B2NEWA       FYY2000    R  +2  REFFLD(FYY *SRC)
B2NEWA       TYY2000    R  +1  REFFLD(TYY *SRC)
```

The date fields in the converted dictionary have been renamed as follows:

- DTINVOICE is called B20000001
- DTPROM is called DTPROM2000
- FYY is called FYY2000
- TYY is called TYY2000

After the conversion, the converted file F1ALIAS contains references that map the date fields to the corresponding fields in the converted dictionary:

```
F1ALIAS
     A                                     REF(DCTALIAS)
     A   R F1ALIASR          TEXT('DATE')
     A
     A
B2MODA     DTINVOICE  R REFFLD(BP20000001)
B2MODA     DTPROM     R REFFLD(DTPROM2000)
B2MODA     FYY        R REFFLD(FYY2000)
B2MODA     TYY        R REFFLD(TYY2000)
```

Appendix A. BYPASS2000 Conversion Repository

The Conversion Repository is made up of about 80 relational tables that can be accessed though native AS/400 query and SQL functions. Each of these tables contains information regarding specific areas for all programs in the application that is to be converted.

For example, the file ANDATFLD contains information related to the fields defined in Copybooks and programs.

The tables in the repository follow the naming convention `XX YYY ZZZ`.

- XX identifies a process.
- YYY identifies a type of activity.
- ZZZ briefly identifies content.

The following table lists possible values for XX and YYY.

Table 7. Values for processes and types of activities

Type	Identifier	Description
Process	AN	Analysis
	CV	Conversion
	HS	List of sources
	US	Working tables
	XW	Working tables
Activity	DAT	COPY and program analysis
	SQL	SQL/DB2 databases
	ERR	Generic use
	TAB	Parameter table

The repository also contains some DATAARAs with their related tables BPPARM and CVPARM. Each table in the repository contains a brief description of the content, which can be accessed with the native AS/400 tools.

Relationships between Repository Tables and Processing Phases

The following sections list which tables in the Conversion Repository are populated after each BYPASS2000 processing phase.

Note: It is very important to display the conversion log, and check for requested information after each processing step to find out if there are any unresolved issues that need to be addressed before proceeding to the next phase.

Table 8. Tables updated while creating conversion environment

Table	Description
HSDATLIB	Tables of old and new libraries
HSDATPAR	Parameter table
HSDATSDT	System-field inventory
HSDATTYF	Date type

Table 9. Tables updated while loading information about application databases

Table	Description
ANDATFFD	File field description
ANDATDBR	Database relationship
ANDATDBK	Database keys
HSDATSRC	Source file list
ANDATFMT	Database keys
HSDATROS	Object/source relationships
CVERRLOG	Warnings generated during analysis and conversion activities

Table 10. Tables updated during memory-level analysis of database information

Table	Description
ANDATFLD	Field storage definition
HSDATSRC	Source file inventory
HSDATFLS	File inventory
HSDATFIL	File/COPY relationship
HSDATLCK	Locked areas
HSDATFLD	Date field inventory
CVERRLOG	All messages generated during analysis and conversion

Table 11. Tables updated during memory-level analysis of COPYs

Table	Description
ANDATFLD	Field storage definition
ANDATVAL	Default values of variables
HSDATLCK	Locked areas
HSDATSRC	Source inventory
ANDATSQI	Exec SQL definition (SQL)
CVERRLOG	Warnings issued during analysis and conversion activities

Table 12. Tables updated during memory-level analysis of SQL source

Table	Description
ANDATTDF	Table definition
ANDATRVT	Relationship between view and table
ANDATFLD	Field definition
ANDATFIL	File with date type
HSDATFIL	File list
HSDATLCK	Locked areas
HSDATSQL SQL	SQL inventory
CVERRLOG	Warning issued during analysis and conversion activities

Table 13. Tables updated during memory-level analysis of programs

Table	**Description**
HSDATINQ	Requested inquiry
HSDATSRC	Source inventory
ANDATRPC	Relationship between program and COPY
ANDATFLD	Field definition
ANDATVAL	Variable default value
ANDATSQI	SQL statement
ANDATCST	Code statement
ANDATRPF	Relationship between program and file
ANDATPRO	Basic relationship between variables
ANDATQST	SQL statement
USDATCCR	Caller-Called relationships
ANDATRNM	File field renamed in program
ANDATRFF	Relationship between INT/EXT file field name
ANDATRFA	Relationship between file and I/O area
ANDATNOE	Areas not to be expanded
USDATPNE	Support for areas not to be expanded
ANDATNOP	Areas not to be propagated
ANDATCCR	Relationship between caller and called programs
ANDATVRC	Relationship between physical file and virtual COPY
CVERRLOG	Messages issued during analysis and conversion activities

Table 14. Tables updated during date-field assignment

Table	Content
HSDATFLD	Date field inventory
HSDATDFI	Assignment imported from external tool
HSDATSDT	General assignment
ANDATFIL	File with date type
HSDATFIL	File list
HSDATLCK	Locked areas
USDATNOE	Support for areas not to be expanded
HSDATTYD	Default for management typology
HSDATFN	External date-field assignment imported into BYPASS2000

Table 15. Tables updated during date-field propagation

Table	**Description**
ANDATOLR	List of discovered date fields in programs
ANDATIPT	Inter-program propagation trace
ANDATOLC	List of discovered date fields defined in Copybooks

Table 15. Tables updated during date-field propagation (continued)

Table	Description
ANDATPTR	Propagation trace
ANDATNOE	Areas not to be expanded
ANDATNOP	Areas not to be propagated
ANDATDNA	Images of areas related to date fields
ANDATDFR	Origin of year-sensitive fields found
ANDATFLD	Field storage definition
ANDATFIL	File with date field
HSDATINQ	Requested inquiry
CVERRLOG	Messages issued during analysis and conversion activities
ANDATOLP	List of date fields that were discovered in program linkage parameters

Table 16. Tables updated during conversion

Table	Description
CVDATRPT	Report definition
CVDATRPL	Report log
ANDATFLD	Field storage definition
CVERRLOG	Messages issued during analysis and conversion activities
ANDATDIZ	New long-date ref-field in dictionary (reference file)
HSDATFRN	Program file names
HSDATSFL	Check flag for a source

Table 17. Table updated by user

Table	Description
HSUSROPT	User-defined options

Table 18. Tables containing source list

Source	Source List Name	Source Type
COPY	HSDATSRC	CPY
Program	HSDATSRC	PGM
DDL SQL	HSDATSQL	DDS
DDS	HSDATSRC	DDS

Appendix B. HSDATDFI Interface File (V3R1M1) Layout

Use the file HSDATDFI to import external field assignment into an environment that is compatible with BYPASS2000 V3R1M1 environment. You can create this file manually, or through a tool such as SEARCH2000. The file must have the format shown in Table 19.

When creating the file manually, make sure that you identify all the date fields that are contained in the physical files of your application. Do not identify fields as dates if you are unsure whether they truly are dates. It is safer to give BYPASS2000 less information than more.

Should you fail to identify a date, the propagation analysis most often detects the date and points out the error. If you identify a date that is invalid, you can cause additional problems.

Table 19. Layout of HSDATDFI

File name: HSDATDFI Record length: 135			
Name	**Type**	**Length**	**Description**
FLDNAME	CHAR	30	Field Name
PARENTNAME	CHAR	30	Must be blank.
IOAREANAME	CHAR	30	I/O-area name This is the name of the data structure or record format of which this field forms a part. See “I/O Area Name” on page 185 for more information.
SRCNAME	CHAR	10	Source-member name BYPASS2000 assumes that the combination of member name and source type is unique in the application. See “Source Member Name” on page 185 for more information.
SRCTYPE	CHAR	3	Source type DDS, CPY, or PGM See “Source Member Type” on page 185 for more information.
DTFTYPE	ZONED	3,0	See “Date Field Type” on page 185 for a list of recognized types.

Table 19. Layout of HSDATDFI (continued)

File name: HSDATDFI Record length: 135			
Name	**Type**	**Length**	**Description**
DTYLEN	ZONED	1,0	Length of the year portion of the date. Valid values are 2 and 4.
EXPTYPE . -	CHAR	1	Expansion type: • 0 = field to be expanded • 1 = field NOT to be expanded See “Expansion Type” on page 186 for more information.
FILTYPE	CHAR	4	File type See “File Type and File Name” on page 186 for more information.
FILNAME	CHAR	10	File name See “File Type and File Name” on page 186 for more information.
FLGLOCK	CHAR	1	Area to be locked: • ’Y’ = Yes • ’N’ = No The usual setting is ’N’. See “Area to be Locked” on page 186 for more information.
PRPTYPE	CHAR	1	Propagation type: • ’0’ = Propagate this field • ’1’ = Do not propagate this field See “Propagation Type” on page 186 for more information.
FLDSSTPOS	ZONED	5,0	Starting position of the substring within the field. Normally 1. See “Substring Start Posn and Length” on page 186 for more information.

Table 19. Layout of HSDATDFI (continued)

File name: HSDATDFI Record length: 135			
Name	**Type**	**Length**	**Description**
FLDSSTLEN	ZONED	5,0	Length of the substring. Normally equal to the field length. See “Substring Start Posn and Length” on page 186 for more information.
FLGELAB	CHAR	1	For internal use only. Set to ‘N’

I/O Area Name

For file fields, the I/O Area Name should be the record format name preceded by an asterisk (*). If the field is contained in a data structure, the I/O Area Name is the name of that data structure. If the field does not relate to any data structure or record format, this field should be set to the same value as the field name.

Source Member Name

You specify the actual libraries and source files when you create the conversion environment. Although you may specify multiple source files in each category, BYPASS2000 assumes that the combination of Source member name and Source type are unique. If the application to be converted contains multiple members with the same name, you must remedy the situation before BYPASS2000 analyzes the source.

SEARCH2000 should warn you if it encounters such a situation.

Source Member Type

Do not confuse this with the regular SEU type. The source-member type indicates whether the member represents a program (PGM), file (DDS), or COPY member (CPY). COPY members include those that are used to provide a file description where files are internally described.

Date Field Type

If you encounter dates that are not in this list, you should identify them as type 001 (Year) and make the appropriate entries in the Substring position and Substring length fields (FLDSSTPOS and FLDSSTLEN).

```
Code  Short   Description
      ----  -----   -----------
      001   Year    Year
      002   YMD     Year,Month,Day
      003   DMY     Day,Month,Year or Month,Day,Year
      004   D-M-Y   Day-Month-Year or Month-Day-Year WITH SEPARATORS
      005   Y-JUL   Year and Julian day
      006   MY      Month,Year
      007   YM      Year,Month
      010   Cent.   Century (e.g. the CC portion of a CCYYMMDD date)
```

The following types may appear in older applications (particularly those derived from mainframe code). C- does not indicate a Century digit. It indicates complement dates.

```
011   C-Yea   Complementary Year
012   C-YMD   Complementary Year,Month,Day
015   C-JUL   Complementary Year,Julian day
017   C-YM    Complementary Year,Month
```

Expansion Type

The normal action of BYPASS2000 is not to extend any dates that it encounters in display files, printer files, or O specs. For example, these dates would normally have an EXPTYPE of 1. You can override this default on a field-by-field basis. If you use an external tool to create the HSDATDFI file, it is up to the tool vendor to provide some means to control this value. Otherwise you can make changes to specific fields through the BYPASS2000 interface.

File Type and File Name

Only fields in physical files, programs, and COPYs can be assigned. For programs or COPYs you must put spaces in these fields. For physical files you must set the File Type field to PF and the File Name to the name of the file. Any logical files that use the physical files automatically receive their date-field assignment.

Area to be Locked

It is unlikely that you will want to set this to anything other than N. If this field is set to Y, BYPASS2000 interprets that this field never contains a date. It is strongly recommended that the lock option is not used by any tools that build the HSDATDFI file.

Propagation Type

This option determines that if this field touches another field, that field is automatically considered a date. Normal operation is to propagate the field; therefore, the value should be 0.

Substring Start Posn and Length

If the whole field is a date, the position (FLDSSTPOS) should be set to 1, and the length (FLDSSTLEN) set to the length of the field. If the date is contained within a larger field, the position is set to the offset within the field, and the length to the length of the date.

For example, if an invoice number occupies a single database field but contains a year and a month (it is in the format nnnnnYYMM), there are two options:

- Assign the full date.

  DFTYPE is 007, FLDSSTPOS is 6, and FLDSSTLEN is 4.
- Assign only the year.

  DFTYPE is 001, FLDSSTPOS is 6, and FLDSSTLEN is 2.

If the format of the date portion is nnnnnMMYY, the options are:

- Assign the full date.

  DFTYPE is 006, FLDSSTPOS is 6, and FLDSSTLEN is 4.
- Assign only the year.

  DFTYPE is 001, FLDSSTPOS is 8, and FLDSSTLEN is 2.

Once BYPASS2000 has completed the memory-level analysis, you can import assignment information from the interface file. To do this you should follow the steps described in “Importing Date-Field Assignment from HSDATDFI” on page 117.

Appendix C. HSDATDFN Interface File (V3R1M2) Layout

Use the HSDATDFN file to import external field assignment into an environment that is compatible with BYPASS2000 V3R1M2. You can create this file manually, or by using option 13 (Load field assignment into HSDATDFN for export) on the Date-Field Assignment menu. The file has the same format as the HSDATDFI file described in Table 19 on page 183, with the following additional entries:

Table 20. Layout of additional entries in HSDATDFN

File name: HSDATDFN Record length: 135			
Name	**Type**	**Length**	**Description**
IOAREAPRG	ZONED	3,0	I/O-area sequence See “IOAREAPRG” for more information.
FLDTYPE	CHAR	1	Field type See “FLDTYPE” for more information.
DTYOFFSET	ZONED	5,0	Year position See “DTYOFFSET” for more information.

IOAREAPRG

If BYPASS2000 finds incongruent REDEFINES clauses in COBOL, it can generate different I/O areas with the same I/O-area name, distinguished by different sequence numbers. Use IOAREAPRG, to select one of these I/O areas specifically.

If there is only a single I/O area with that name, you must set IOAREAPRG to 0. If there are more than one I/O areas with the same name, you must specify a valid sequence number for IOAREAPRG.

FLDTYPE

Specify the type (binary, packed, zoned or char) of the assigned area. Generally, you specify the same type as for the assigned field.

However, you can assign a date with a type that is different from the field type. For example if you have a char field, you can assign a packed substring from position 1 and with length 2.

DTYOFFSET

For zoned or char date fields, the value of this field must always be 1. For binary or packed date fields, specify the position of the year, in digits, in the packed or binary area.

Consider the following examples:

```
JULDAT    PIC   S9(5) COMP-3   seeded as a Julian date
Y D D
Y D F       dtyoffset = 1
```

and

```
DMYDAT    PIC   S9(6) COMP-3   seeded as DMY
- D M Y
D M Y F     dtyoffset = 6
```

Note: If the date type is 010 or 020, DTYOFFSET is the position of the century or the century flag.

Once BYPASS2000 has completed the memory-level analysis, you can inport assignment information from the interface. To do this you should follow the steps described in “Importing Date Field Assignment from HSDATDFN” on page 118.

Appendix D. Markers Added to Converted Sources

The following markers are reported in the HSDATPAR file in the *xxxx*DB library.

RPG	COBOL	Description	Intervention Required
B2???	BP2???	Identifies instructions that cannot be converted by BYPASS2000 and marks the existence of an incongruence between the two areas involved in the instruction. Probably one instruction contains date fields, while the other cannot contain date fields because of previously found limitations.	YES
B2000	BP2000	Marks an instruction that involves areas or date fields but does not need to be changed. The lengthening of the year-related information in both data fields involved is sufficient to handle century data. This marker can also refer to instructions that involve figurative constants.	NO
B2CHK	BP2CHK	Marks potential conceptual problems in the conversion of the instruction. BYPASS2000 has determined that the instruction involves date fields. You must verify the logic of the converted code.	YES
B2CPY	BP2CPY	Marks the point at which copies have been inserted in the program. The Copybooks contain the instructions for adding and removing the century or the century flag, and the relevant work areas.	NO
B2MOD	BP2MOD	Marks instructions that modify the data definition.	NO
B2NEW	BP2NEW	Marks lines of codes that have been added for adding/removing century information.	NO
B2OLD	BP2OLD	Marks the lines of code that have been commented out and replaced by new code to add/remove century information.	NO
B2REM		Marks comments that signal the beginning and end of new lines of code and information for highlighted conversion problems.	NO
B2TRC	BP2TRC	Marks instructions in which normal truncation of the numeric field type is used to remove century information.	NO
B2VFY	BP2VFY	Marks instructions that BYPASS2000 has changed based on parameters shown in BPPARM, or based on the type of operation. You must verify these instructions.	YES
B2INF	BP2INF	Marks information for the user, for example, after a B2CHK.	YES
B2LEA	BP2LEA	Marks an instruction that involves a leap-year calculation.	YES
B2SHF	BP2SHF	Marks lines of code that have been added for shifting century information, and field definitions with shifted positions.	NO
	BP2OUT	Marks lines of code in which a year-sensitive field is right-truncated in a non-year-sensitive field.	YES

RPG	COBOL	Description	Intervention Required
BP2CEN	BP2CEN	Marks an instruction that involves a field with a single century digit or a single century-flag digit.	NO
	BP2CUT	Marks an instruction that involves a date field and a non-date field for which the option ″Delete relationship between storage areas″ was used. You must verify the logic of the converted code.	YES
	B2SST	Marks an instruction that involves substrings that are not handled during conversion.	YES

Glossary

Analysis Phase. Includes all operations that record information about the sources and objects of an application.

Anchor Field. A field that has been declared to contain year-sensitive data in a specific format. During the data propagation phase, anchor fields are used as starting points to identify related year-sensitive fields.

Assignment of date fields. The operation that declares fields as being year-sensitive and specifies their format. It must be performed on file layouts, but can also be performed on program areas.

Conversion. The operation that produces sources that can handle 4-digit YEAR information, for an application that could only handle 2-digit YEAR information.

Conversion Database (or Repository). The database that contains the information about an application gathered from user-input (during conversion-environment creation and field assignment), and from analysis results.

Data Integrity Module (DIM) Dispatcher. A program called by the converted application. It takes as input the name of a file to check, and calls the appropriate Date-Integrity Module (DIM) program.

Data Integrity Module (DIM) Programs. Programs that check the logical and formal integrity of date fields when they are written to the database of a converted application.

Generic field. A field for which BYPASS2000 cannot determine whether it contains date data or not, or what the format of the date data should be. Generic fields are either not reached during the propagation analysis phase, and therefore their status cannot be determined, or they contain data of more than one type, and BYPASS2000 needs additional information to determine their format.

Incongruence. An incongruence is a relationship between two fields. If a field receives both date information from a year-sensitive field, and non-date information from a non-year-sensitive field, BYPASS2000 considers this an incongruent relationship and flags the field as a generic area. Incongruences require user intervention to manually assign and lock the generic areas.

I/O Area. I/O areas are memory locations that correspond to named or unnamed data structures.

I/O Area Sequence Number. BYPASS2000 assigns sequence numbers as a way of differentiating I/O areas that correspond to unnamed data structures.

Locked Field. A field that has been declared not to contain year-sensitive data. During the data propagation phase, locked fields are used as starting points to identify related non-year-sensitive fields.

Memory-Level Analysis. The function that identifies the memory areas that correspond to the data items of an application, and populates the conversion repository.

Migration Programs. The programs that generate the converted database that can handle 4-digit YEAR information, starting from the original database that could only handle 2-digit YEAR information.

Original Source. The source of the application to be converted. Note that the original source is not changed during the conversion. Only for COBOL/400 BYPASS2000 inserts a normalization line in the code.

Propagation Analysis. The operation that identifies date fields, referred to in the application source. It is based on user-defined anchor fields, and information gathered during memory analysis.

Repository. See conversion database.

Seeding. See assignment of date fields.

Test Programs (or Time-Shift Programs). Programs that perform a time shift of the database by a user-defined number of years, to test the behaviour of a converted application before, at, and after the turn of the 21st century.

Index

Special Characters

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

W

Y

Program Number: 5697–D11

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC09-2591-01