

Table of Contents

1	Introducing CHART	5
1.1	What CHART does.....	5
1.2	When to use CHART.....	7
1.3	Invoking CHART On-line.....	8
1.4	CHART, the Year 2000, and NATURAL Versions.....	9
1.5	CHART in German, French, Spanish, and Italian.....	9
2	Release Notes for CHART Version 3.1F	10
3	Defining what you want to CHART	14
3.1	Choosing a Seed.....	15
3.2	Excluding Objects.....	15
3.3	Trace of CHART being built.....	17
3.4	Don't Explode / Don't Show / Only Explode Objects.....	18
3.5	Options for drawing 'Inline Subroutines' and 'Database Files'.....	19
3.6	CMWKF01, CMWKF02, or CMWKF04 as "Don't Explodes".....	19
3.7	Order of Searching.....	20
3.8	Step Libraries.....	20
3.9	Batch Invokers.....	21
3.10	PF Keys on the CHART "Seed" Menu.....	21
3.11	NATURAL Statements that CHART looks for.....	22
3.12	External Subroutines should be CATALOGed.....	23
3.13	CHART Manual On-line.....	24
4	Formats, Symbols, and Abbreviations	25
4.1	Large Screens (27x132, 32x80, 43x80).....	25
4.2	Changing connectors, colours, and date formats.....	26
4.3	Changing the "Inverse" (^) Character.....	27
4.4	Changing the language CHART is displayed in.....	27
4.5	Automatically saving CHART parameters.....	28
4.6	Date, Time, Seed, Page, User, and Library in CHART Headings.....	29
4.7	Calling Method.....	30

4.8	Objects called dynamically.....	31
4.9	Objects called more than once.....	31
4.10	Don't Explode Objects.....	31
4.11	Object with no source code.....	32
4.12	Objects whose descendants could not be placed.....	32
4.13	"Long" Names for Inline Subroutines and Database Files.....	32
5	PF Keys when viewing a CHART.....	33
5.1	Help with a CHART (PF1).....	33
5.2	Browsing source code (PF2).....	34
5.3	Zooming in and out on the CHART (PF4).....	35
5.4	Finding objects in a CHART (PF5).....	36
5.5	Drawing a new CHART (PF6).....	37
5.6	Paging around the CHART (PF7, PF8, PF10, and PF11).....	38
5.7	Calling Method and Line, Object Length, and Save Details (PF9).....	38
6	Browsing Source code from within a CHART.....	41
6.1	Commands and PF Keys available while Browsing.....	43
6.2	Scanning for text while Browsing.....	44
6.3	Browse any object (without leaving Browse).....	47
7	Unplaced Objects within a CHART.....	48
7.1	How unplaced objects are indicated.....	48
7.2	What to do when there are unplaced objects.....	49
7.3	Unplaced objects in batch CHART-ing.....	51
8	How to tell CHART to draw Batch Invocation.....	52
9	Drawing CHARTs in Batch.....	54
10	High Volume CHART-ing in Batch.....	57
10.1	The seed list, CMWKF01, and the 'table of contents'.....	57
10.2	Wildcard in the seed name.....	58
10.3	CMWKF03, CMWKF04, and the 'index of objects'.....	58
10.4	Page limit and the 'not referenced' list.....	58
10.5	Sample JCL for high-volume CHART-ing.....	59
10.6	Less redundancy with special Don't explode values.....	64

10.7	Example CHARTs using special Don't explode values.....	66
10.8	Changing excluded object types and Don't explode objects	71
10.9	Changing the Natural Library and Step Libraries	72
10.10	Potential NAT1205 (No more sort work space) Error	73
10.11	Details of CHART-ed objects held in CSV Format on CMWKF04	76
10.12	Importing CHART-ed objects into Visio2000.....	78
11	Messages used in CHART	80
12	Installing CHART	82
12.1	Create target dataset	82
12.2	Copy "binary" software file to target datasets	82
12.3	Set up some "Natload" JCL.....	83
12.4	Run the "Natload" Job.....	85
12.5	Copy Required SAG subprograms from SYSEXT.....	86
12.6	Make CHART a steplib for relevant users/applications	86
12.7	Enter your License Information.....	87
12.8	Access to the default FUSER	87
13	Customising CHART	88
13.1	Remembering previous CHART parameters.....	88
13.2	Access to source code in other FUSER files.....	89
13.3	Order of searching FUSER files (and other repositories).....	89
13.4	Default "Object Types" to be Excluded.....	89
13.5	Default "Trace Interval"	90
13.6	Default "Don't Explode" / "Don't Show" / "Only Explode" Settings.....	90
13.7	Current Library and Step Libraries.....	90
13.7.1	Initialising Step Libraries using option 'USR1025N'	90
13.7.2	Initialising Step Libraries using option 'NSCLI'.....	91
13.7.3	Initialising Step Libraries using option 'NONATSEC'	91
13.7.4	Initialising Step Libraries using option 'AUTOSTEP'.....	91
13.7.5	Initialising Step Libraries directly in 'CHART'	91
13.8	Default "Batch Invocation" Parameters.....	92
13.9	Default Settings for CHART characters and colours	92
13.10	CHART and NATURAL Versions	92
14	How to get CHART -ed.....	93

Harvest Moon
Computing

CHART

2605 Nicholson Road, Suite 230
Sewickley, PA 15143
Phone: 724.759.7070
Fax: 724.759.7067

Pty Ltd abn 23 054 256 914

Version 3.1F

www.treehouse.com

Index.....94

Harvest Moon
Computing

Pty Ltd abn 23 054 256 914

CHART

Version 3.1F

2605 Nicholson Road, Suite 230
Sewickley, PA 15143
Phone: 724.759.7070
Fax: 724.759.7067
www.treehouse.com

1.3 Invoking CHART On-line

Invoke *CHART*, from the Command or NEXT prompts, by typing “ch” or “chart”.

For example, at the NEXT prompt ...

```
NEXT chart                                LIB=ANYLIB
```

or at the Command prompt ...

```
Enter Code:      Type:      Name:
Command: ch
```

The initial *CHART* “seed” menu will then be displayed ...

```
      B  Debugging Facility
      L  List Objects, X-Ref
      G  Global Environment
      Draw New Chart with New Criteria ...
Seed .. _____ Exclude .. DSCO_____ Trace 5000_
Don't .. ^M*_____ ^DB*_____
Expl/Show _____
Command ==>
Command =
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help About Quit Main Order StpLb Batch Excln Shape      Doco Parm
```

Alternatively, the *CHART* parameters (seed name, exclusions, trace interval, etc.) can be entered directly after the *CHART* command, and the “seed” menu will not be displayed ...

```
Enter Code:      Type:      Name:
Command: CH PROG1 DSC
```

(The fields and parameters for the initial *CHART* menu are explained in the following section.)

1.4 CHART, the Year 2000, and NATURAL Versions

CHART is fully Year 2000 compliant and fully compatible with all versions of NATURAL up to and including Version 3.1.2.

The only date manipulation in *CHART* is to display the current date and time and to display the last saved date of a NATURAL object, when requested. While the century has always been available for display in previous versions of *CHART*, with *Version 3.1F* the century is now included in date displays. The format of dates within *CHART* can be changed at any time. The order of day, month, and year can be chosen as well as the character which separates them. The default format is 'YMD' separated by '-', eg. 2001-02-18.

CHART Version 3.1F has been developed in NATURAL 2.2.8 but also operates correctly in previous versions of NATURAL as well as NATURAL Version 2.3 and Version 3.1. Some SAG supplied 'USR' subprograms need to be available to *CHART*. See '12.5 Copy Required SAG subprograms from SYSEXT' on Page 86 of this manual for details.

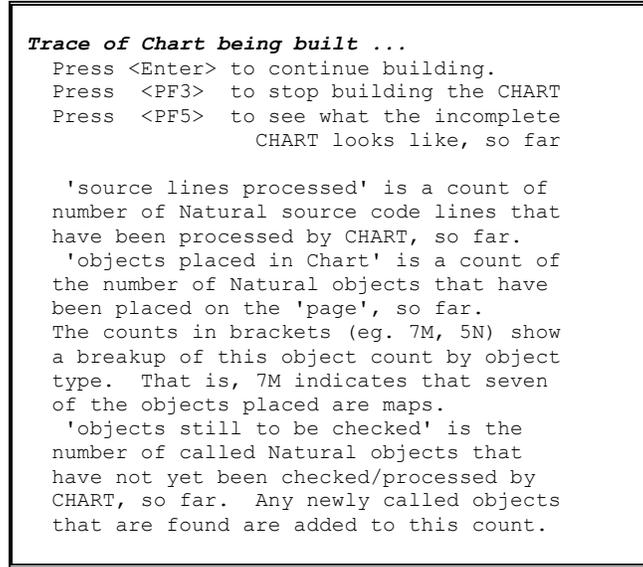
While *CHART Version 3.1F* operates correctly under NATURAL Version 2.2.8, official support of *CHART* for NATURAL Version 2.2.8 (and all versions prior to Version 2.2.8) will be discontinued after December 31, 2001.

1.5 CHART in German, French, Spanish, and Italian

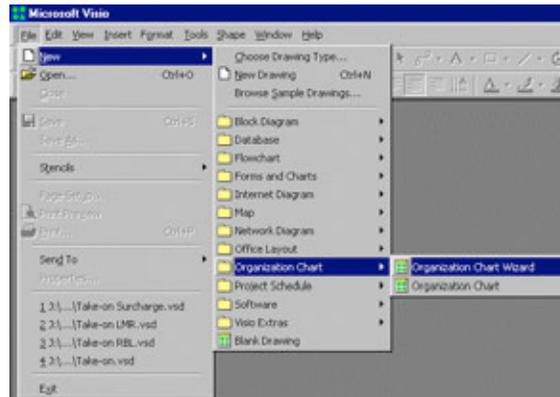
CHART is now available in German, French, Spanish, and Italian (as well as English). All screens, messages, and help can be viewed in these languages. To access this feature, use the %L=n terminal command, as follows:

%L=1	will give you	English
%L=2	will give you	German
%L=3	will give you	French
%L=4	will give you	Spanish
%L=5	will give you	Italian

- *CHART Version 3.1F* has extra help screens to fully explain the use of PF keys;



- *CHART Version 3.1F* now produces a work file that can be read into Microsoft Visio2000. Information about every object drawn by *CHART* is automatically written to CMWKF04 in batch and can be read directly into Visio2000 as organisation charts;



- *CHART* is now available in German, French, Spanish, and Italian. All screens, messages, and help can be viewed in these languages. To access this feature, use the %L=n terminal command, where 1 = English, 2 = German, 3 = French, 4 = Spanish, and 5 = Italian;
- While *CHART Version 3.1F* operates correctly under NATURAL Version 2.2.8, official support of *CHART* for NATURAL Version 2.2.8 (and all versions prior to Version 2.2.8) will be discontinued after December 31, 2001.

3 Defining what you want to CHART

CHART needs (at least) the name of the **seed** or “initial” object from which it will draw a **program chart**. Other parameters and criteria can also be provided on the “seed” menu. Certain object types can be **excluded** from the generated chart. For example, you may want to **exclude** all maps and data-areas from the chart, depending on your objective. Also, you can specify which objects are NOT to be **exploded** in the chart.

Enter command ‘ch’ or ‘chart’ and the **window-size** “seed” menu will be displayed ...

```

      B   Debugging Facility
      L   List Objects, X-Ref
      G   Global Environment

      Draw New Chart with New Criteria ...
Seed .. _____ Exclude .. DSCO_____ Trace 5000_
Don't .. ^M*_____ ^DB*_____
Expl/Show _____
Command ==>

Command =
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help About Quit Main Order StpLb Batch Excln Shape      Doco ParmS
  
```

Press **PF9** to change the ‘shape’ of this menu to **full-screen** ...

```

17:44:48          **** CHART V3.1F for Natural ****          2001-02-18

User HMC01      Code  Function                               Library DEVLIB
-----
      G   Generate Chart
      L   License Details
      ?   Chart Help
      .   Terminate Chart
-----
Code .. G      Seed .. MYPROG__

Exclude .. GO_____ M Map           W Work Files   C Copycode
      B Batch Calls  X Xtrnl Calls  K Set Key     P Program
      R Routines    (S Subroutine N Subprogram H Helproutine )
      D Data Areas  (G Global Area L Local Area A Param Area )
      O Other Types (F D'base File I Inline Subr )

Trace 5000_ source lines (or Page Limit in 'Batch')

Don't .. Expl/Show X*_____ Z*_____
      _____
      _____
      _____
      _____
Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help About Quit Main Order StpLb Batch Excln Shape      Doco ParmS
  
```

Apart from the field called “Code”, all other parameters and PF keys are identical between the **window-size** and **full-screen** “seed” menus.

Code: Type “G” for “Generate Chart”.

Exclude Objects: Press PF9 on the *CHART* "seed" menu to see the 'Exclusions' window. The object types currently excluded will be shown. Mark the object types that you want excluded:

```

User HMC01          - Development Facilities

                                Code  Function
                                ----  -
                                C    Create Object
                                E    Edit Object
                                R    Rename Object
                                D    Delete Object
                                X    Execute Program
                                T    DB Command Log Facility
                                B    Debugging Facility
                                L    List Objects, X-Ref
                                G    Global Environment

                                Draw Chart with Criteria
                                Seed .. MYPROG__ Exclude ..
                                Don't .. X*_____ Z*_____
                                Expl/Show _____
                                _____
                                Command ==>

Command =
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF
Help About Quit Main Order StpLb Batch Ex

                                Exclusions
                                _ M Map
                                _ D Data Areas
                                X G Global Area
                                _ L Local Area
                                _ A Param Area
                                _ P Program
                                _ K Set Key
                                _ R Routines
                                _ S Subroutine
                                _ N Subprogram
                                _ H Helproutine
                                _ C Copycode
                                _ X Xtrnl Calls
                                _ B Batch Calls
                                _ W Work Files
                                X O Other Types
                                _ I Inline Subr
                                _ F D'base File

                                MDHWF InputOutput
                                _ DPKSNXB Prog. Calls
  
```

3.3 Trace of CHART being built

On the *CHART* “seed” menu, there is an option called ‘Trace.’ with a default value of 5000. This means that a **trace** screen will be displayed for every 5000 lines of source code processed. This **trace** screen shows the total number of source lines processed and the number of objects placed (so far) summarised by object type (eg. 7P means seven programs have been placed, 5N means five subprograms, and so on).

```
Trace of Chart being built ...

5000 source lines processed
23 objects placed in Chart
(7P 5N 9M 2L)
6 objects still to be checked

--Enter-Continue--PF5-ShowChart--PF3-Exit--
```

You may continue with the generation of the chart by pressing **Enter**, or to go back to the “seed” menu by pressing **PF3**. To see the (incomplete) *CHART* at this point, press **PF5** (ShowChart). You can change the default of 5000 lines on the “seed” menu. To STOP the **trace** screen being displayed, use a value of zero (0).

Press **PF1** (Help) on this screen and the following help screen will be displayed:

```
Trace of Chart being built ...
Press <Enter> to continue building.
Press <PF3> to stop building the CHART
Press <PF5> to see what the incomplete
          CHART looks like, so far

'source lines processed' is a count of
number of Natural source code lines that
have been processed by CHART, so far.
'objects placed in Chart' is a count of
the number of Natural objects that have
been placed on the 'page', so far.
The counts in brackets (eg. 7M, 5N) show
a breakup of this object count by object
type. That is, 7M indicates that seven
of the objects placed are maps.
'objects still to be checked' is the
number of called Natural objects that
have not yet been checked/processed by
CHART, so far. Any newly called objects
that are found are added to this count.
```

3.4 Don't Explode / Don't Show / Only Explode Objects

Many applications have **utility** objects that are called from almost every other object. (For example, generalised error or menu handlers.) You can tell *CHART* not to **explode** or not to **show** such objects, as they add little to the meaning of the *CHART*. Up to twenty **don't explode** / **don't show** / **only explode** objects (or groups of objects) can be specified.

Don't Explode ... objects will be drawn in the chart (highlighted with a "+") but will never have descendants. For example, if 'MYPROG2' uses some copycode called 'ERRORHDL' and a **don't explode** of 'ERR*' was used, then the *CHART* would look like this:



Don't Show ... objects will not be shown in the *CHART* at all. A *CHART* of 'MYPROG2' with a **don't show** of '^ERR*' would just show 'MYPROG2'. The copycode 'ERRORHDL' is not shown at all:



Only Explode ... objects will be shown and exploded in the *CHART*, but all other object will be shown but **not** exploded, eg. '~MYP*'.

Only Show ... objects will be shown and exploded in the *CHART*, but all other object will not be shown **at all**, eg. '-MYP*'.

Groups of objects can be specified using wildcard notation: "*", "!", and ")" :

- M* objects **starting with "M"** will have descendants suppressed.
- M!!C* objects starting with "M", with **anything in position 2 and 3**, with "C" in position 4, anything after that will have descendants suppressed.
- M!!C*)PN objects starting with "M", with anything in position 2 and 3, with "C" in position 4, and anything after that, and that are **objects of type "program" or "subprogram"** will have descendants suppressed.
- ^M* objects starting with "M" will **not be shown in the CHART at all**.
- XMC* Only Explode objects starting with "XMC", Don't Show other objects at all.
- ~XMC* Only Explode objects starting with "XMC", Show other objects, but Don't Explode them.

3.5 Options for drawing 'Inline Subroutines' and 'Database Files'

```

You have NOT excluded 'Inline Subroutines' or 'Database Files'.
You have a choice about how these object types are drawn ...

. . . . .

EITHER as 'parents' of other objects ...
      CAPTURE1
      |
      +-----+
      |         |         |
f:READ- i:CHECK- i:AMEND-
  DATA  DATA   DATA
  |
i:CHECK-
  DATA

      InputMap
      AMENDM1

OR only as 'children' of another
Natural object, never 'parents' ...

      CAPTURE1
      |
      +-----+
      |         |         |         |
i:CHECK- f:READ- i:AMEND- InputMap
  DATA  DATA   DATA   AMENDM1

To get this format, enter 'don't
explodes' of 'INLINESUBRTN'
and/or 'DATABASEFILE'

. . . . .

Don't Explode 'INLINESUBRTN'? _ Don't Explode 'DATABASEFILE'? _
    
```

You can decide how **Inline Subroutines** and **Database Files** are drawn in a *CHART*. They can be drawn either as “child” or “parent” objects. If you haven’t excluded them, **Inline Subroutines** and **Database Files** will be drawn as “parent” objects by default. If you use the special **Don’t Explode** values of **INLINESUBRTN** and/or **DATABASEFILE** then they will be drawn only as “child” objects.

3.6 CMWKF01, CMWKF02, or CMWKF04 as “Don’t Explodes”

For “high volume” *CHART*-ing in batch, you can specify that every other object in **CMWKF01**, **CMWKF02**, or **CMWKF04** is to be treated as a **Don’t Explode** (or **Don’t Show**) object, automatically. For example, if an object is about to be “exploded” in a *CHART* but that object is also in **CMWKF01**, then the object’s children will NOT be exploded. Special **Don’t Explode/Show** values of **CMWKF01**, **CMWKF02**, **CMWKF04**, **^CMWKF01**, or **^CMWKF02** are used to invoke this feature.

See section 10.6 **Less redundancy with special Don’t explode values** for more information.

3.7 Order of Searching

Press PF5 on the *CHART* “seed” menu to view (and alter) the ‘order of searching’. This order is important for clients with more than one location for source code, eg. multiple FUSER files, source code in archiving products like NMCF, or source code on several platforms (MVS and UNIX).

```

User HMC01          - Development Facilities -          Library DEVLTR
                                Define Order of Searching
                                Order of Searching repositories ... F_____
Code -----
C      Subprograms 'HMCINFOx', 'HMCLINEx', and 'HMCSUBRx'
E      are used to access Natural application source code
R      repositories, where 'x' is usually one of ...
D      F: ADABAS 'FUSER' File (Mainframe)
X      G: General access via 'USR1057N' (M/F or UNIX)
T      N: NMCF (Natural Migration Control Facility)
B      U: UNIX Directory ('C' Programs)
L      W: Windows Directory ('C' Programs)
G

The source code of these subprograms are provided
and may be amended and copied to suit your needs.

For example, if you have two 'FUSER' files to hold
your source code, copy the 'HMC...F' subprograms
to 'HMC...P' and amend the DDMS to point to the
other 'FUSER' file. Then 'P' could be included
in the 'Order of Searching' specified above.

Seed ..
Don't ..
Expl/Show
Command ==

Enter-PF1---PF2---PF3---P
      Help About Quit Main Order StpLb Batch Excln Shape      Doco ParmS
  
```

3.8 Step Libraries

Press PF6 on the *CHART* menu to see this window. The step libraries are initialised by *CHART*. The way they are initialised is chosen when *CHART* is installed. The most common method is to use those step libraries defined in NATURAL SECURITY. Amend the libraries as you require:

```

Step Libraries ... DEVL_____
                  TEST_____
                  PROD_____

< to be searched  _____
in order after   _____
CURRLIB >       _____
                  _____
                  _____
                  _____
  
```

3.9 Batch Invokers

Press PF7 on the *CHART* “seed” menu to view (and alter) the current **batch invoker** definitions. Batch invocations of NATURAL using the standard Software AG program NATRJE are automatically detected and drawn. Some existing clients (who use “homegrown” pre-cursors to NATRJE) have had these “batch invokers” incorporated directly into the *CHART* software. For ease of use, up to four simple “batch invokers” can be defined in the initial *CHART* program and/or by using PF7 on the “seed” menu.

```

X   Execute Program
T   DB Command Log Facilit
B   Debugging Facility
L   List Objects, X-Ref
G   Global Environment
    Draw New Chart with N
Seed .. _____ Exclude
Don't .. ^M* _____ ^DB* _____
Expl/Show _____
Command ==>>

Batch Parameters ...
Object which   Natural program
invokes batch  to be run in
Natural ...    batch is in
               Parameter ...

1. NATSUB_    3_
2. _____
3. _____
4. _____

Command =
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---
Help About Quit Main Order StpLb Batch

```

3.10 PF Keys on the CHART “Seed” Menu

The following is a summary of the PF keys available on the “seed” menu:

	Chart	General	Help
PF1	Help	Displays these help screens.	
PF2	About	Displays CHART license information.	
PF3	Quit	Exits from the CHART 'Entry screen'.	
PF4	Main	Exits from the CHART 'Entry screen'.	
PF5	Order	Allows the order of searching to be changed if more than one source code repository is used, e.g. two FUSER files.	
PF6	StpLb	Allows the step libraries to be changed. CHART will look for NATURAL objects in the current library first, and will then look in each of the step libraries.	
PF7	Batch	Allows 'batch invokers' to be defined. These are the program(s) in your environment that are called online to submit a batch job.	
PF8	Excln	Allows you to choose which object types are to be excluded from the CHART.	
PF9	Shape	Changes the shape of the CHART entry screen.	
PF11	Doco	Displays CHART Documentation on-line.	
PF12	Parms	Allows various CHART parameters to be changed.	

3.11 NATURAL Statements that CHART looks for

CHART looks for the following NATURAL statements in your source code (depending on which object types you may have excluded), identifies the object called/invoked by the statement, and draws that object along with a descriptive “calling method”:

<u>Called Using NATURAL Statement</u>	<u>Calling Method</u>	<u>Dynamic Call</u>	<u>Relates To Object Type</u>
CALL	Call		External
CALL	Call#	Yes	External
CALLNAT	CallNat		Subprogram
CALLNAT	CallNat#	Yes	Subprogram
DELETE FROM	DbseFile, f:.....		<several>
FETCH	Fetch		Program
FETCH	Fetch#	Yes	Program
FETCH RETURN	FetchRet		Program
FETCH RETURN	FtchRet#	Yes	Program
FIND	DbseFile, f:.....		<several>
GET	DbseFile, f:.....		<several>
GLOBAL USING	GlobalDA		Global Data Area
HISTOGRAM	DbseFile, f:.....		<several>
INCLUDE	Include		Copycode
INPUT USING MAP	InputMap		Map
INPUT USING MAP	InptMap#	Yes	Map
INSERT INTO	DbseFile, f:.....		<several>
LOCAL USING	LocalDA		Local Data Area
PARAMETER USING	ParamDA		Parameter Data Area
PERFORM	Perf Ext		External Subroutine
PERFORM	Perf Int, i:.....		Inline Subroutine
PROCESS SQL	DbseFile, f:.....		<several>
READ	DbseFile, f:.....		<several>
READ WORK FILE	ReadWork		<several>
RUN	Run		Program
RUN	Run#	Yes	Program
SELECT FROM	DbseFile, f:.....		<several>
SET KEY	SetKey		Program
SET KEY	SetKey#	Yes	Program
STACK COMMAND	StackCom		Program / Command
STACK COMMAND	StckCom#	Yes	Program / Command
USING HELP	HelpRout		HelpRoutine
USING HELP	HlpRout#	Yes	HelpRoutine
WRITE USING FORM	WrteForm		Form
WRITE USING FORM	WrteFrm#	Yes	Form
WRITE WORK FILE	WrteWork		<several>

3.12 External Subroutines should be CATALOGed

CHART reads the source code of Natural objects and finds which other objects are called. When *CHART* finds a call to an external subroutine, some extra non-source code information is needed. This is because the actual name of the external subroutine can be different from the name of the object that it is stored in. For example, the following subroutine is called JANE but defines a subroutine called FRED. The subroutine is invoked with “PERFORM FRED”:

```
Object JANE:  
  
DEFINE SUBROUTINE FRED  
*  
WRITE 'HERE IS FRED'  
*  
END
```

The link between the name of the object JANE and the subroutine name FRED is recorded in the FUSER file when the object is STOWed or CATALOGed. When *CHART* sees the source code “PERFORM FRED” (and “FRED” is not an inline subroutine), the link information (ie. FRED is defined in object JANE) is retrieved from the FUSER file and *CHART* then reads the source code of JANE. This is the only time that *CHART* uses FUSER information other than source code.

For the above reasons, it is recommended that external subroutines be CATALOGed before *CHART* is used.

3.13 CHART Manual On-line

Press PF11 on the CHART “seed” menu to view this manual on-line. Select the chapter(s) and section(s) that you want to view.

Press PF7 to page up within a section and PF8 to page down.

Press PF3 to leave a section or chapter.

```
2002-02-04          Manual_for_CHART31F          TBF1
Select a Chapter with any character.
- 1 Introducing CHART
- 2 Release Notes for CHART Version 3.1F
- 3 Defining what you want to CHART
- 4 Formats, Symbols, and Abbreviations
- 5 PF Keys when viewing a CHART
- 6 Browsing Source code from within CHART
- 7 Unplaced Objects within a CHART
- 8 Tell CHART to draw Batch Invocation
- 9 Drawing CHARTs in Batch
- 10 High Volume CHART-ing in Batch
- 11 Messages used in CHART
- 12 Installing CHART
- 13 Customising CHART
- 14 How to get CHART -ed

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Exit
```

4 Formats, Symbols, and Abbreviations

4.1 Large Screens (27x132, 32x80, 43x80)

CHART Version 3.1F supports large screen displays. While the normal 24x80 mainframe screen (also known as IBM 3279 Model 2) is commonly used in NATURAL applications, many terminal emulators now support the larger screen sizes: 32x80 (IBM 3279 Model 3), 43x80 (IBM 3279 Model 4), and 27x132 (IBM 3279 Model 5). CHART now automatically detects the use of the larger screen displays and adjusts its own display to show you as much as possible. The following is an example of the 27x132 (IBM 3279 Model 5) screen:

```

HMC0029: Press PF12 to view Objects with Descendants NOT Drawn.
2001-05-09 09:04 ~ C ~ ~ ~ D ~ ~ ~ E ~ ~ PRFINIT_ ~ G ~ ~ ~ H ~ ~ ~ I ~ ~ ~ J ~ ~ ~ K ~ ~ ~ L ~ ~ ~ M ~ ~ Page: 1L

b |-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
LocalDA LocalDA LocalDA LocalDA LocalDA CallNat CallNat Fetch CallNat CallNat Call Call StackCom Fet
PRFVCTRL PRFVCUPL PRFVHDL PRFVUSRL PRFVUGLL PRFTEXT1 USR0050N PRFUE99F USR2004N USR0011N PROFINIT PROFCLOS SETUP PRFO

c |-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
FetchRet Fetch LocalDA CallNat WriteForm Fetch
PRFSAVE PRF2100B PRFVHDL PRFTEXT1 PRF000B1 PRFUE99F

d |-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
StackCom StackCom LocalDA StackCom
PRF6000P PRF0000P PRFVUGLL PRF5200P

e |-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
CallNat LocalDA LocalDA Call Fetch Fetch Fetch Fetch Fetch Fetch Fetch Fetch Fetch Fetch
PRFTEXT1 PRFVHDL PRFVCKL PROFINIT PRF3500B PRF2300B PRF7340B PRF3100B PRF7550B PRF3200B PRF2200B PRF7100B PRF7

f |-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
LocalDA LocalDA LocalDA CallNat Fetch
PRFVCKL PRFVCTRL PRFVHDL PRFTEXT1 PRFUE99F

g |-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
PRF600R1 PRF0000P PRF60011 PRFUE99P PRF600L1 PRF6400P PRF6300P PROFINIT PRF1000P PRFUE99P PRF4000P PRF5200P PRF0000P PRF6
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help Brwse Quit Zoom Find Chart Up Down Lines Left Right Ovrfl

```


4.3 Changing the “Inverse” (^) Character

Press PF12 on the CHART “seed” menu to view the current display settings ...

```

Amend Chart Display Settings...

      AD= CD= Connector      Automatically Save Parameters? Y
Seed V  RE  Down |         Your last CHART params. will be saved for you
Connector D  YE  Cross _
Outline D  NE  Inverse
Warning B  YE  Value ^
Date Order YMD Separate? Y with Separator - ==> 01-02-18 and 2001-02-18
: _ A _ _ _ B _ _ _ C _ _ _ D _ _ _ E _ _ :
Call.Mthd Obj.Name      2001-02-18 12:38 SEED Page: 1 of 1 :
AD= CD= AD= CD=      :
Map I  PI  V  PI      b | _____ | _____ | _____ | b
Global Area I  NE  V  NE      CallNat GlobalDA InputMap LocalDA Fetch
Local Area I  NE  V  NE      NNNNNNNN GGGGGGGG MMMMMMMM LLLLLLLL PPPPPPPP
Param Area I  NE  V  NE      : | _____ | _____ | _____ | :
Program I  GR  V  GR      c | _____ | ? | _____ | c
Subroutine I  BL  V  BL      Batch ParamDA HelpRout StackCom Include
Subprogram I  BL  V  BL      PPPPPPPP AAAAAAAA HHHHHHHH PPPPPPPP CCCCCCCC
Helproutine I  BL  V  BL      : | _____ | _____ | _____ | :
Copycode D  TU  V  TU      d | _____ | _____ | _____ | d
Inline Subr I  BL  V  BL      ReadWork WrteWork Call Perf Int DbseFile
Work Files I  PI  V  PI      WWWWWWWW WWWWWWWW PPPPPPPP IIIIIIII FFFFFFFF
D'base File I  NE  V  NE      : _ A _ _ _ B _ _ _ C _ _ _ D _ _ _ E _ _ :

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Quit      Updte      Let
  
```

To specify a “Don’t Show” object (rather than a “Don’t Explode” object), you must use an “inverse” character before the object name, eg. ^HM*. By default, the inverse character is the carat symbol (^). However, some sites do not use this character on keyboards and/or video displays. CHART allows you to change the inverse character to one suitable to your environment. Enter the character that you require in the “Inverse Value” field, and press PF6 to update your change. This character can be changed at any time.

4.4 Changing the language CHART is displayed in

CHART is now available in German, French, Spanish, and Italian (as well as English). All screens, messages, and help can be viewed in these languages. To access this feature, use the %L=n terminal command, as follows:

- %L=1 will give you **English**
- %L=2 will give you **German**
- %L=3 will give you **French**
- %L=4 will give you **Spanish**
- %L=5 will give you **Italian**

4.5 Automatically saving CHART parameters

Press PF12 on the CHART “seed” menu to view the current display settings ...

```

Amend Chart Display Settings...

      AD= CD= Connector      Automatically Save Parameters? Y
Seed V  RE  Down |         Your last CHART params. will be saved for you
Connector D  YE  Cross _
Outline D  NE  Inverse
Warning B  YE  Value ^
Date Order YMD Separate? Y with Separator - ==> 01-02-18 and 2001-02-18
      : _ A _ _ _ B _ _ _ C _ _ _ D _ _ _ E _ _ :
Call.Mthd Obj.Name         2001-02-18 12:38 SEED Page: 1 of 1 :
      AD= CD= AD= CD=      :
Map I  PI  V  PI          b | _____ | _____ | _____ | b
Global Area I  NE  V  NE  CallNat GlobalDA InputMap LocalDA Fetch
Local Area I  NE  V  NE  NNNNNNNN GGGGGGGG MMMMMMMM LLLLLLLL PPPPPPPP
Param Area I  NE  V  NE  : | _____ | _____ | _____ | :
Program I  GR  V  GR      c | _____ | ? _____ | c
Subroutine I  BL  V  BL   Batch ParamDA HelpRout StackCom Include
Subprogram I  BL  V  BL   PPPPPPPP AAAAAAAA HHHHHHHH PPPPPPPP CCCCCCCC
Helproutine I  BL  V  BL   : | _____ | _____ | _____ | :
Copycode D  TU  V  TU     d | _____ | _____ | _____ | d
Inline Subr I  BL  V  BL   ReadWork WrteWork Call Perf Int DbseFile
Work Files I  PI  V  PI   WWWWWWWW WWWWWWWW PPPPPPPP IIIIIIII FFFFFFFF
D'base File I  NE  V  NE   : _ A _ _ _ B _ _ _ C _ _ _ D _ _ _ E _ _ :

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Quit      Updte      Let

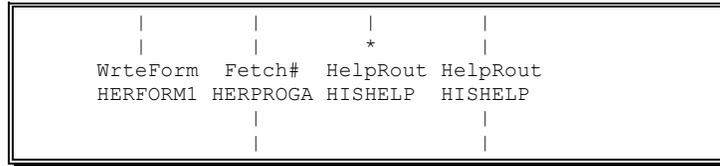
```

With CHART, each user’s most recent CHART parameters can be ‘remembered’. That is, whichever seed name, object exclusions, don’t explodes, etc. were last used (on-line), the next time that user invokes CHART (on-line), those parameters will be re-displayed as the default settings.

The field ‘Automatically Save Parameters?’ controls how this feature works. By default, this field is set to ‘Y’ which means each user’s parameters will be saved each time they use CHART.

If a particular user does NOT want their parameters saved in this way, the user should change ‘Automatically Save Parameters?’ to ‘N’, and then press PF6 (Updte). This process can be reversed at any time.

4.7 Calling Method



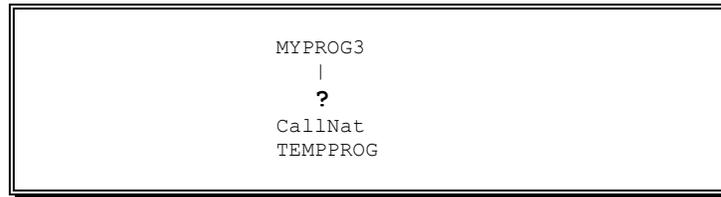
Each object displayed in the chart has its **calling method** shown immediately above it, such as: InputMap, CallNat, FetchRet.

<u>Calling Method</u>	<u>Called Using NATURAL Statement</u>	<u>Dynamic Call</u>	<u>Relates To Object Type</u>	<u>On-Line Colour</u>
Batch	<call to batch NATURAL program>		<several>	turquoise
Call	CALL		External	turquoise
Call#	CALL	Yes	External	turquoise
CallNat	CALLNAT		Subprogram	blue
CallNat#	CALLNAT	Yes	Subprogram	blue
DbseFile	READ, FIND, GET, HISTOGRAM, SELECT FROM, DELETE FROM,		<several>	white
f:.....	READ, FIND, GET, HISTOGRAM, SELECT FROM, DELETE FROM,		INSERT INTO, PROCESS SQL <several>	white
Fetch	FETCH		Program	green
Fetch#	FETCH	Yes	Program	green
FetchRet	FETCH RETURN		Program	green
FtchRet#	FETCH RETURN	Yes	Program	green
GlobalDA	... GLOBAL USING		Global Data Area	white
HelpRout	... USING HELP		HelpRoutine	blue
HlpRout#	... USING HELP	Yes	HelpRoutine	blue
i:.....	PERFORM		Inline Subroutine	blue
Include	INCLUDE		Copycode	turquoise
InputMap	INPUT USING MAP		Map	pink
InptMap#	INPUT USING MAP	Yes	Map	pink
LocalDA	... LOCAL USING		Local Data Area	white
ParamDA	... PARAMETER USING		Parameter Data Area	white
ReadWork	READ WORK FILE		<several>	pink
Run	RUN		Program	green
Run#	RUN	Yes	Program	green
Perf Ext	PERFORM		External Subroutine	blue
Perf Int	PERFORM		Inline Subroutine	blue
SetKey	SET KEY		Program	green
SetKey#	SET KEY	Yes	Program	green
StackCom	STACK COMMAND		Program / Command	green
StckCom#	STACK COMMAND	Yes	Program / Command	green
WrteForm	WRITE USING FORM		Form	pink
WrteFrm#	WRITE USING FORM	Yes	Form	pink
WrteWork	WRITE WORK FILE		<several>	pink

4.11 Object with no source code

? (Source Code Not Found)

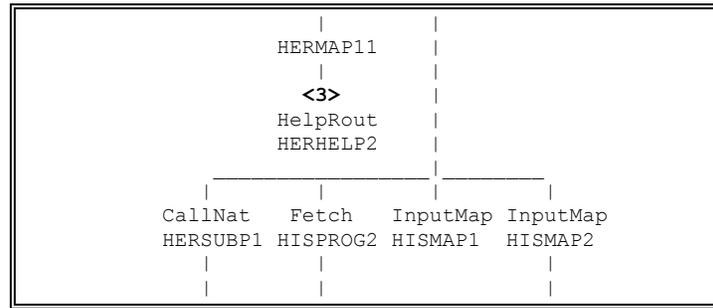
Sometimes the source code for an object is **not in the current** library (or any of the Step Libraries, if specified). This is indicated with a “?” when the object is drawn in the chart. Its descendants can not be drawn.



4.12 Objects whose descendants could not be placed

<n> (Descendants could not be placed)

For large complex *CHARTs*, some descendant objects cannot be placed, normally because others “branches” of the *CHART* are “in the way”. (See ‘7 Unplaced Objects within a *CHART*’ on Page 48.) When this happens, the number of descendant objects are shown within angle brackets (eg. “<3>”) is shown above the object whose descendants could not be placed. Objects with descendants that could not be placed (if any) can be viewed by pressing PF12 on-line. In batch, extra *CHARTs* will be drawn automatically for each “parent” object which had descendants that could not be placed.



4.13 “Long” Names for Inline Subroutines and Database Files

Most objects drawn by *CHART* have names that are a maximum of eight characters long. Inline Subroutines and Database Files can have names up to 32 characters long. Names longer than 14 characters long are shortened by removing embedded vowels (ie. ‘a’, ‘e’, ‘i’, ‘o’, ‘u’) starting from the end of the name (until 14 characters long). The first 14 characters of the name are always drawn in the *CHART*. If there is room below this, the remainder of the name is drawn.

The full name of the Inline Subroutine or Database File can be viewed using the PF5 (Find) key. This will display a list of all objects in the *CHART* in alphabetical order.

5 PF Keys when viewing a CHART

5.1 Help with a CHART (PF1)

Context sensitive help is available throughout the *CHART* software. **Help (PF1)** on a *CHART* will display the following screen:

```
Chart 'Display' Screen Help
```

Key	Command	Effect
---	-----	-----
PF2 *	Browse	Browse Object Source
PF3	Quit	Back to Entry Screen
PF4	Zoom	Zoom in/out on Chart
PF5 *	Find	Find Object in Chart
PF6 *	Chart	Draw a new Chart
PF7	Up	Back one Page
PF8	Down	Forward one Page
PF9	Lines/Lngth/SvUsr/SvDte/Libry..	...
	...change display above object name	
PF10	Left	Show Left of Chart
PF11	Right	Show Right of Chart
PF12	Overflow	Show Unplaced Objects

(* means key is cursor sensitive,
ie. move cursor to object you
require and press PF key)

Column **ids** for the Chart (eg. A, B, C)
are shown **above** or **below** each **column**.
Row **ids** for the Chart (eg. a, b, c) are
shown to the **left** or **right** of each **row**.

5.4 Finding objects in a CHART (PF5)

Find (PF5) allows you to Find all occurrences (if any) of an object within a *CHART*. By pressing **PF5**, a small window will be shown at the bottom of the screen. All objects in the *CHART* are shown in **alphabetical order** on this window and **full “paging”** facilities are provided.

```

2001-02-18 16:23  - C - - - D - - - E - - - F - - - HMCTEST  - H - - - I -
d |-----|-----|-----|-----|-----|-----|-----|-----|
i:EDT-TF i:EDT-EN CallNat DbseFile i:VRFY-P DbseFile CallNat i:EDT-
N-AND-RK TTY-NME- TFXCHNGE PYR-DTLS MT-RFRRL PYR-NM HMCNINDC DR-AND
Y-IF-INV BRTH-DTE NTCT
e |-----|-----|-----|-----|-----|-----|-----|-----|
CallNat CallNa
HAZ Find Object.. PYR-DTLS ACXCHNGE HAZNST
f |-----|-----|-----|-----|-----|-----|-----|-----|
Row Col Objects in Alphabetical Order
d D PAYER-DETAILS
d F PAYER-NAME
h I RCBPAM72
g |-----|-----|-----|-----|-----|-----|-----|-----|
d M RCCCAM62
h J RCCCAM63
g Seed: HMC
|-----|-----|-----|-----|-----|-----|-----|-----|
DbseFile Call
NUMBER RCBPAM
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help Quit Find Top Up Down Bot
  
```

You can page through the list of objects, or **type** the **name** of the object you want to find in the field provided. Press **enter** or **PF5** again and all the row/column references at which the object appears (if any) will be reported in the message line.

```

HMC0028: Object PYR-DTLS found at row/column: dD dN gJ
2001-02-18 16:23  - C - - - D - - - E - - - F - - - HMCTEST  - H - - - I -
d |-----|-----|-----|-----|-----|-----|-----|-----|
i:EDT-TF i:EDT-EN CallNat DbseFile i:VRFY-P DbseFile CallNat i:EDT-
N-AND-RK TTY-NME- TFXCHNGE PYR-DTLS MT-RFRRL PYR-NM HMCNINDC DR-AND
Y-IF-INV BRTH-DTE NTCT
e |-----|-----|-----|-----|-----|-----|-----|-----|
CallNat CallNa
HAZ Find Object.. PYR-DTLS ACXCHNGE HAZNST
f |-----|-----|-----|-----|-----|-----|-----|-----|
Row Col Objects in Alphabetical Order
d D PAYER-DETAILS
d F PAYER-NAME
h I RCBPAM72
g |-----|-----|-----|-----|-----|-----|-----|-----|
d M RCCCAM62
h J RCCCAM63
g Seed: HMC
|-----|-----|-----|-----|-----|-----|-----|-----|
DbseFile Call
NUMBER RCBPAM
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help Quit Find Top Up Down Bot
  
```


6.1 Commands and PF Keys available while Browsing

There are many commands available while “browsing” source code. The “browse” commands are similar in style to those used in the normal mainframe NATURAL object editor. Commands can be entered in a field at the top left of the “browse” window. Most commands can also be invoked using a PF Key.

HELP	(PF1):	Display a Help window describing all the commands.
QUIT	(PF3):	Stop browsing source code and return to the <i>CHART</i> .
SC, SCAN	(PF4):	Scan for text. See section ‘6.2 Scanning for text while Browsing’ on Page 44.
+P,SC=	(PF5):	Go down one Page and Scan again.
KEYS/NOKEY	(PF6):	Show or Don’t Show the current PF Key settings at the bottom of the screen.
-P	(PF7):	Go up one Page.
+P	(Enter, PF8):	Go down one Page.
SHAPE	(PF9):	Changes the Shape of the browse window. By default, the Shape is a full screen. Other options are “wide”, “tall”, and “quarter”.
LEFT	(PF10):	Show the Left hand side of the source code lines.
RIGHT	(PF11):	Show the Right hand side of the source code lines.
SHIFT	(PF12):	Shifts the browse window to a different position over the <i>CHART</i> . For example, from the bottom half to the top half.
TOP,SC=	(PF17):	Go to the Top of the object and Scan again.
-H	(PF19):	Go up Half a page.
+H	(PF20):	Go down Half a page.
TOP	(PF22):	Go to the Top of the object.
BOT	(PF23):	Go to the Bottom of the object.
-nnnn:		Go up “nnnn” lines.
+nnnn:		Go down “nnnn” lines.
nnnn:		Go to line “nnnn”.

The text to be scanned for can be entered directly after the SCAN command, eg. SC END-DEFINE. If just 'SCAN' (or 'SC') is entered, a window will appear allowing the text to be entered or a previously scanned-text to be selected.

```

2001-02-18 12:41  - C - -  QR1000  - E - - - F - - - G - - Page: 1L of 1
                                     |                                     |
b +-----+-----+-----+-----+-----+-----+-----+-----+
Fetch LocalDA Fetch Perf Ext CallNat InputMap
XXXFER ZJOBL ZJOB QR1000 DATERNGE QR1000M1
                                     |

```

```

> scan
Brws  ....+....1....+....2
3110 END-DECIDE /*(2990)
3120 *
3130 DECIDE ON FIRST VALU
3140 VALUE 'S'
3150 MOVE TRUE
3160 MOVE ' ??' L
3170 MOVE ' 11'
3180 MOVE 'PORT'
3190 VALUE 'D'
3200 MOVE TRUE
Browse ....+... PF1=? PF3=Q

```

```

Scan for Text ...
Select -----
Previous -----
Scan ----- END-DEFINE
Values ----- #FUNCTION
----- VIEW OF
----- Description
OR ----- FETCH
----- INPUT USING
New Value... DECIDE _____
Absolute ... X

```


6.3 Browse any object (without leaving Browse)

CHART allows you to browse any object (within your current step libraries) from within the “browse” feature of the CHART display screen. While browsing an object, type ‘BR’ or ‘BROWSE’ and an object name in the command line and you will immediately begin browsing the nominated object:

```

2001-02-18 12:41  - C - -  QR1000  - E - - - F - - - G - - Page: 1L of 1
                                     |                                     DEVLIB  Lib.
b + -----+-----+-----+-----+-----+-----+-----+-----+
Fetch  LocalDA                Fetch  Perf Ext CallNat  InputMap
XXXXFER  ZJOBL                ZJOB   QR1000  DATERNGE  QR1000M1

```

```

                                     Browse Object ...
> browse nx4000                                Program  QR1000  Lib  DEVLIB
Brws  .....1.....2.....3.....4.....5.... Mode Struct
3110 END-DECIDE /* (2990)
3120 *
3130 DECIDE ON FIRST VALUE OF #FUNCTION
3140  VALUE 'S'
3150  MOVE TRUE                                TO #STATS-REPORT-REQD
3160  MOVE ' ??'  L'                          TO #ZJOB-SYSOUT-LIST
3170  MOVE ' 11'                                TO #ZJOB-COPIES-LIST
3180  MOVE 'PORT'                              TO #ZJOB-USER-ID
3190  VALUE 'D'
3200  MOVE TRUE                                TO #DETAILED-REPT-REQD
Browse ..... PF1=? PF3=Q PF5=+19,SC= PF7=-P PF8=+P ...5... S 344  L 311

```


7.2 What to do when there are unplaced objects

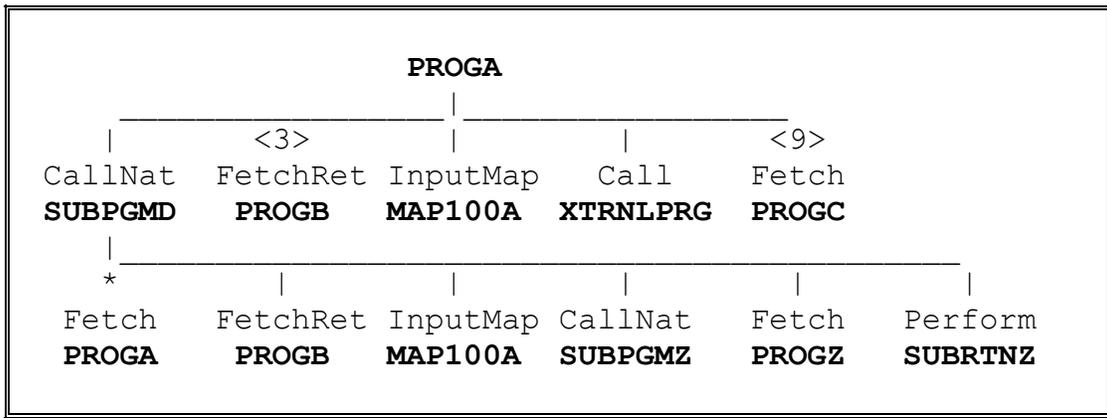
The 15 row x 15 column limit allows one ‘chart’ to be completely printed on a standard mainframe page (ie. 132 print characters x 60 print rows). A 15 x 15 chart can be shown on-line with minimal scrolling.

The 15 x 15 limit also means that a maximum of 211 objects can be shown on one ‘page’ (15 x 15 = 225, less 14 because only the ‘seed’ is shown on the first row). It is extremely unlikely that 211 objects or calls will fit into the 15 x 15 matrix because many of the available 211 ‘slots’ will get blocked off by various branches of the ‘chart’.

If you were drawing the ‘chart’ by hand, unplaced objects mean roughly the same as ‘I don’t have any more room on my piece of paper’. You would probably decide to:

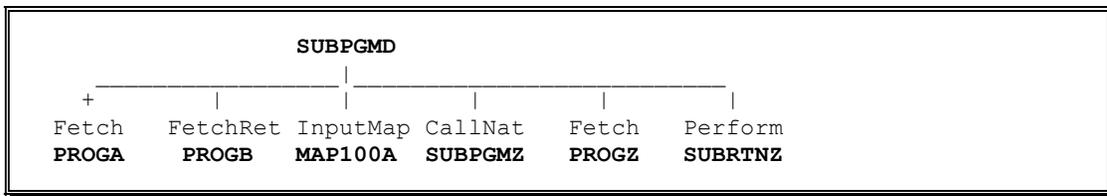
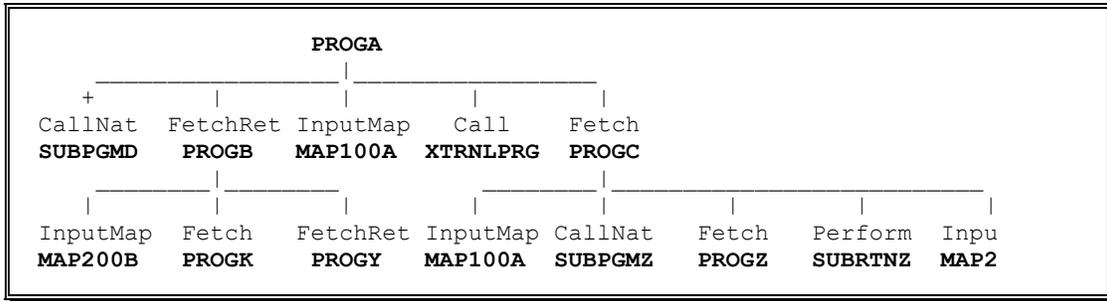
- split the ‘chart’ into two or more logical pieces, with ‘connector’ symbols pointing to the various pieces; or
- reduce the level of detail in your hand drawn ‘chart’ -- make room on your page by erasing less important parts of the ‘chart’.

With *CHART* you can do the same by use of “don’t explodes” and object type “exclusions”. For example, say we draw a ‘chart’ with a seed called ‘PROGA’:



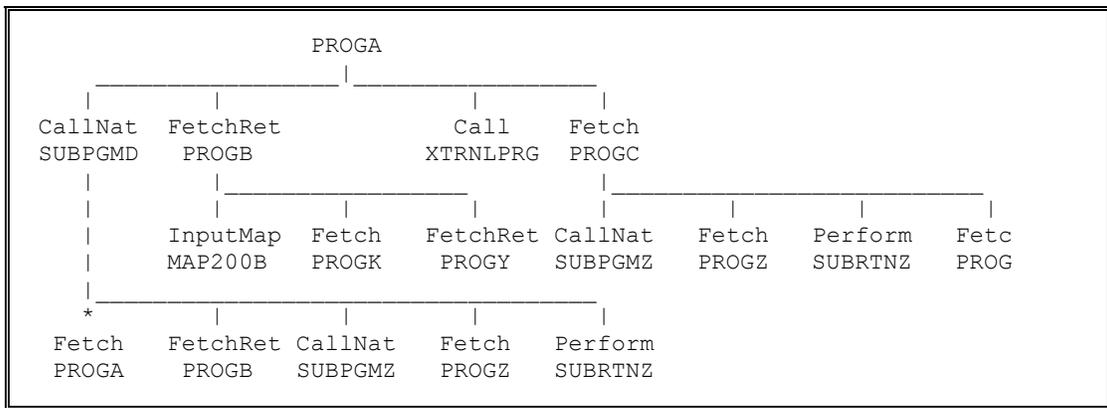
The 'chart' is too crowded and two objects ('PROGB' and 'PROGC') have three (<3>) and five (<5>) child objects respectively which cannot be drawn. Either split the 'chart' in two or reduce the level of detail, as follows.

- Make 'SUBPGMD' a "don't explode" for the original 'chart' and then draw a new 'chart' for 'SUBPGMD' (with 'PROGA' as a "don't explode"):



Or ...

- add maps (M) to your object type "exclusions" and re-draw the 'chart':



In general:

- consider "excluding" object types which are not important to your current purpose (HelpRoutines, WorkFiles, InlineSubroutines, and DatabaseFiles are good candidates).
- consider using more "don't explodes" and "don't shows" (standard 'error handlers', 'menu handlers', GDAs, LDAs are good candidates).

- if you are using the ‘seed list’ (CMWKF01) in batch, consider using the special “don’t explode” values: ‘CMWKF01’, ‘CMWKF02’ or ‘CMWKF04’.

7.3 Unplaced objects in batch CHART-ing

In batch, extra *CHARTs* are automatically drawn for each object that has “unplaced” child objects. That is, if the first *CHART* drawn has any “unplaced” objects, an extra *CHART* will be drawn automatically for each different “unplaced” parent within the first *CHART*.

A “table of contents” and “index of objects” covering all the resulting *CHARTs* will also be created automatically. If you don’t want extra *CHARTs* to be drawn automatically, specify a “page limit” of one (1) immediately after the seed and immediately before any object type exclusions.

If you are using the ‘seed list’ (CMWKF01) in batch, the number of unplaced objects can be reduced by using the special “don’t explode” values: ‘CMWKF01’, ‘CMWKF02’, or ‘CMWKF04’.

For example, if you use ‘CMWKF01’ as one of your “don’t explodes” then all seeds in the ‘seed list’ (ie. in file CMWKF01) will be treated as “don’t explodes” within every ‘chart’:

eg. CH CMWKF01, DSCO, 200, ^ERR*, CMWKF01, MENU*

If you use ‘CMWKF02’ as one of your “don’t explodes” then all the seeds in the ‘original seeds plus extra seeds’ (ie. in file CMWKF02) will be treated as “don’t explodes” within every ‘chart’.

eg. CHART CMWKF01, GO, 0, CMWKF02, MENU*

If you use ‘CMWKF04’ as one of your “don’t explodes” then all the objects in the ‘details of all objects chart-ed in CSV format’ (ie. in file CMWKF04) will be treated as “don’t explodes” within every (subsequent) ‘chart’. This option stops redundant *CHART-ing* because each object with children will only be “exploded” once. In subsequent pages such an object will still be drawn, but with a ‘+’ to indicate it has NOT been exploded. The page/row/col where the object was originally exploded will also be shown above the object name.

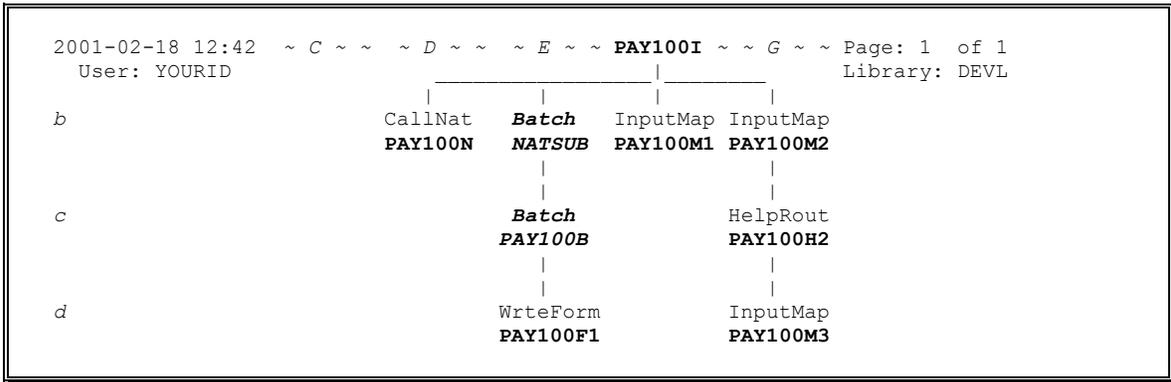
eg. CHART CMWKF01, GO, 0, CMWKF04, MENU*

See the section ‘9 Drawing *CHARTs* in Batch’ on Page 54 for more details.

This is a simple “batch invocation” and can be defined to *CHART* (in the initial ‘CHART’ program) with the following ...

```
MOVE 'NATSUB' TO #BATCH-INVOKER-1
MOVE 03      TO #BATCH-PARM-1
```

CHART will detect the CALLNAT to ‘NATSUB’, will look at the third parameter of the CALLNAT, will determine what value(s) it has been set to, and will draw ...



When using *CHART* on-line, the settings of the four simple “batch invokers” can be viewed (and altered) by pressing PF7 on the “seed” menu ...

```

X Execute Program
T DB Command Log Facilit
B Debugging Facility
L List Objects, X-Ref
G Global Environment

Seed .. Draw New Chart with N
Don't .. ^M* Exclude
Expl/Show
Command ==>

Batch Parameters ...
Object which invokes batch Natural program
Natural ... to be run in
Parameter ... batch is in

1. NATSUB 3
2.
3.
4.

```

9 Drawing CHARTs in Batch

CHART can be used in the **batch** environment, allowing charts to be **printed** on lineprinters or laserprinters for review or documentation. Some sample JCL for running *CHART* in batch follows:

```
//<userid>C JOB 'CHART V2',CLASS=A,MSGCLASS=X,NOTIFY=<userid>
//*
//NATBATCH EXEC PGM=...
//STEPLIB DD DSN=...
// DD DSN=...
//DDCARD DD DSN=...
//*
//CMPRINT DD SYSOUT=*
//CMPRT01 DD SYSOUT=*
//CMPRT02 DD SYSOUT=*
//CMPRT03 DD SYSOUT=*
//CMPRT04 DD SYSOUT=*
//*
//CMWKF01 DD DSN=<seeds identified for extra charts>,DISP=OLD
//CMWKF02 DD DSN=<seeds plus any extra parents>,DISP=OLD
//CMWKF03 DD DSN=<data file for index of objects>,DISP=OLD
//CMWKF04 DD DSN=<details on each object charted>,DISP=OLD
//*
//CMSYNIN DD *
LOGON <library>
GLOBALS IM=D
CHART <seed-1>,<exclude-types>,<page-limit>,<don't-explode>,...
CHART <seed-2>,<exclude-types>,<page-limit>,<don't-explode>,...
...
FIN
/*
//
```

The ‘chart’ commands within the CMSYNIN ddname tell *CHART* what to do. For example, the command:

```
CHART IMP340I1,GO,20,XM*
```

will tell *CHART* to draw a chart with a seed of IMP340I1 using object type exclusions of ‘GO’ (Global Data Areas, Inline Subroutines, and Database Files) with a page limit of 20 and a “don’t explode” of ‘XM*’ (don’t explode any objects starting with ‘XM’).

Any number of charts can be drawn in this way. If more charts are required, simply repeat the line:

```
CHART <seed-n>,<exclude-types>,<page-limit>,<don't-explode>,...
```

and enter the extra seed along with the object exclusions, page limit, and “don’t explodes” you require. An example chart drawn in batch is shown on the next page.

For *CHART*s drawn in batch the current date, time, user, and library are shown at the top of each page. The heading of the *CHART* also shows the seed name, any object type exclusions, any step libraries, and any “don’t explodes” in effect when the *CHART* was drawn.

In batch, object type exclusions are abbreviated in the following way:

Copy	Copycode	GDA	Global Data Area	Spgm	Subprogram
Pgm	Program	LDA	Local Data Area	Help	Helproutine
Xtnl	External Program	PDA	Param Data Area	Isrt	Internal Subroutine
XSrt	External Subroutine	Btch	Batch Call	DbFl	Database File
Skey	Set Key	Work	Work Files		

Row identifiers (a, b, c, ...) and column identifiers (A, B, C, ...) are shown on the boundaries of the *CHART*. If there is room on the page, the calling line number (eg. (5710)) is shown above the calling method (eg. CallNat) and the object’s length (200 lns) is shown below the object name (eg. UZHCONFL).

In **batch**, the generated *CHART*s are written to Natural report file two (CMPRT02). This report file can be directed to any normal system output class or device. The report file can also be directed to a dataset. This dataset should have a record format of **FBA** (Fixed length records, **B**locked, **A**NSI Carriage Control Characters) and a record length of **133**.

Error messages are written to Natural report file zero (CMPRINT) so this report file should also be included in your JCL (as in above example). Report files one (CMPRT01), three (CMPRT03) and four (CMPRT04) should also be included to allow for **table of contents**, **index of objects**, and **not referenced list** respectively.

Not referenced list will only be generated if you specify a “page limit” of zero (0).

Table of contents will not be generated if you specify a “page limit” of one (1).

Work files 1 to 4 should also be defined when running *CHART* in batch. The datasets referred to by these work files should have the following attributes:

<u>DD Name</u>	<u>Record Format</u>	<u>Record Length</u>	<u>Disposition</u>
CMWKF01	FB	80 bytes/characters	Read/Write, OLD/NEW
CMWKF02	FB	80 bytes/characters	Read/Write, OLD/NEW
CMWKF03	FB	20 bytes/characters	Read/Write, OLD/NEW
CMWKF04	FB	160 bytes/characters	Read/Write, OLD/NEW

Blocksize for these datasets can be any multiple of the record length and should conform to your site’s record blocking conventions. Do NOT use a dataset DISPosition of MOD (for MODify) for these work files in your JCL as this will lead to unpredictable results during the generation of the “Index of Objects” and “Table of Contents”. If CMWKF04 is used as special “don’t explode” value, then you must also define CMWKF05 in your JCL, with the same record format and record length as CMWKF04.

10 High Volume *CHART*-ing in Batch

CHART can be used in **batch** to generate a large number of charts in one background job. If all the seeds of your application can be identified, *CHART* can be used in **batch** to completely **document** and **audit** that application, in the following way:

- pass all the **seeds** of your application to *CHART* (in CMWKF01);
- a **table of contents** (including page numbers) will be created (in CMPRT01);
- charts will be generated for each **seed** (in CMPRT02);
- extra charts will be generated for each new **unplaced object** (in CMPRT02);
- an **index** of objects will be created (showing date last saved, library, and every page number / row / column at which the object appears) (in CMPRT03);
- an **audit** list of objects ‘not referenced’ can be created (in CMPRT04).

10.1 The seed list, CMWKF01, and the ‘table of contents’

CHART can read the names of the **seed** objects from a dataset (Natural work file - CMWKF01) and generate a chart for each **seed**. This dataset can also hold a 70 character title for the **seed** object. The seed name and its 70 character **title** will be printed at the head of its chart. The last 2 characters of each 80 character record are reserved for use by the *CHART* software.

An example of records which could be in CMWKF01:

```
IMP340I1 Interim Movement Processing, Part 1
```

CHART will read ‘IMP340I1’ as the **seed** for a chart, and ‘Interim Movement Processing, Part 1’ will be used as the **title** of the chart. The seed name and the title will appear in the **table of contents**.

If this 'high-volume/dataset' feature is used, *CHART* will automatically create a **table of contents** for the seeds, showing the seed object name, its title, and the page number its chart appears on. The **table of contents** is produced in seed name order, in title order, and also in page number order. The **table of contents** is written to Natural report file one (CMPRT01).

If objects cannot be drawn because the chart is too crowded, an extra chart is drawn (automatically) to completely drawn those objects. This is achieved by temporarily adding the ‘parent’ of the affected objects to the list of seeds in CMWKF01 (and its copy in CMWKF02). The extra charts are included in the **table of contents** with the same title as the original seed, but with “(Cont.)” appended. The extra ‘parent’ objects are removed from CMWKF01 when the **table of contents** is generated.

10.2 Wildcard in the seed name

CHART allows a wildcard to be used in the seed name in batch. For example, if a seed of 'UXD*' is used, then every program, subprogram, subroutine, and help routine starting with 'UXD' in the current library will be added to the seed list in CMWKF01 and *CHART*s will be drawn for each one.

10.3 CMWKF03, CMWKF04, and the 'index of objects'

CHART will also automatically create an **index** of all objects appearing in the charts. The **index** is generated from information recorded in CMWKF03 and CMWKF04 and is written to Natural report file three (CMPRT03) and shows:

- the object name,
- the object length (rounded to the nearest 10 lines),
- the (first) library where the object was found,
- all the pages which the object appears on,
- the row and column which the object appears at (for each page),
- the date the object was last saved.

In this way, *CHART* can be used to completely **document** your application. If all the **seeds** relevant to your application can be identified and entered into a dataset, *CHART* can automatically generate program charts for each seed, along with a **table of contents** and an **index** of all objects. As your application changes over time, seeds can be added to or removed from the seed 'dataset'. *CHART* can then be used at any time to completely re-document your application.

10.4 Page limit and the 'not referenced' list

If a "page limit" of zero is specified, a **not referenced** listing will be written to Natural report file four (CMPRT04). To generate this listing, all the objects identified in all the *CHART*s are checked against the current library and all the step libraries. If any other objects are found on these libraries, they are added to the **not referenced** listing. That is, they are objects **NOT** referenced in any program chart. These objects may no longer be required, or may highlight that a seed was not included in the seed 'dataset'.

At the end of processing, 'extra' seeds (ie. unplaced objects) are removed from CMWKF01 but are left in CMWKF02 for information. Information about every object in every *CHART* is written to CMWKF04 in a CSV (Comma Separated Value) format.

10.5 Sample JCL for high-volume CHART-ing

Sample JCL to run the 'high-volume/dataset' feature of *CHART* in batch follows:

```
//<userid>C JOB 'CHART V2',CLASS=A,MSGCLASS=X,NOTIFY=<userid>
//*
//NATBATCH EXEC PGM=...
//STEPLIB DD DSN=...
// DD DSN=...
//DDCARD DD DSN=...
//*
//CMPRINT DD SYSOUT=*
//CMPRT01 DD DSN=<userid>.TABLE.OF.CONTENTS,
// DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(1,1),RLSE),
// UNIT=SYSDA,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=13300)
//CMPRT02 DD DSN=<userid>.PROGRAM.CHARTS,
// DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(10,10),RLSE),
// UNIT=SYSDA,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=13300)
//CMPRT03 DD DSN=<userid>.INDEX.OF.OBJECTS,
// DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(2,2),RLSE),
// UNIT=SYSDA,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=13300)
//CMPRT04 DD DSN=<userid>.SURPLUS.OBJECTS,
// DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(2,2),RLSE),
// UNIT=SYSDA,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=13300)
//*
//CMWKF01 DD DSN=<userid>.ORIGINAL.SEEDS,DISP=OLD
//CMWKF02 DD DSN=<userid>.ORIGINAL.SEEDS.AND.EXTRA.SEEDS,
// DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(1,1),RLSE),
// UNIT=SYSDA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=16000)
//CMWKF03 DD DSN=<userid>.DATA.FOR.INDEX.OF.OBJECTS,
// DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(5,5),RLSE),
// UNIT=SYSDA,DCB=(RECFM=FB,LRECL=20,BLKSIZE=16000)
//CMWKF04 DD DSN=<userid>.CHARTED.OBJECTS.CSV.FORMAT,
// DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(10,10),RLSE),
// UNIT=SYSDA,DCB=(RECFM=FB,LRECL=160,BLKSIZE=16800)
//*
//CMSYNIN DD *
LOGON <library>
GLOBALS IM=D
CHART CMWKF01,<exclude-types>,<page-limit>,<don't-explode>,...
FIN
/*
//
```

To invoke the 'high-volume/dataset' feature of *CHART*, specify 'CMWKF01' as the seed (as in the above example). The seed 'dataset' (in the above example, <userid>.ORIGINAL.SEEDS) should already exist, with a record format of FB and a record length of 80. The first 8 characters of each record should hold a

seed object name, the next 70 characters can hold that seed's description or title, and the last 2 characters are reserved for use by the *CHART* software.

Blocksize for these datasets can be any multiple of the record length and should conform to your site's record blocking conventions. Do NOT use a dataset DISPosition of MOD (for MODify) for work files 1 to 4 in your JCL as this will lead to unpredictable results during the generation of the "Index of Objects" and "Table of Contents". If CMWKF04 is used as special "don't explode" value, then you must also define CMWKF05 in your JCL, with the same record format and record length as CMWKF04.

Examples follow of parts of a high-volume batch *CHART* execution. The first page of the **table of contents** (page order) is shown, followed by the last three **charts** (pages 53 to 55), the first and last pages of the **index of objects** (pages 56 and 63), and finally the **not referenced list** (page 64).

```

2001-02-18 13:06:02          **** CHART V3.1F for Natural ****          Page i
User:  HMC01                (No Page Limit)          Table of Contents - Page Order          Lib:  DEVLIB
-----
Seed  Page  Title                                     | Seed  Page  Title
-----|-----|-----
AC1000  1  Maintain Cases                                     | EP1200  51  View Cases (Cont.)
AC1002  2  Maintain Cases (Cont.)                         | EC1214M1 52  View Cases (Cont.)
AC1001  3  Maintain Cases (Cont.)                         | EH1000  53  View Hearings
AC2100  4  Maintain Cases (Cont.)                         | EP1000  54  View Parties
AC1205M4 5  Maintain Cases (Cont.)                         | EM1000  55  View Members
AC1205M1 6  Maintain Cases (Cont.)                         |
AC2200M3 7  Maintain Cases (Cont.)                         |
XNHLP50 8  Maintain Cases (Cont.)                         |
AC1000M1 9  Maintain Cases (Cont.)                         |
AC1209  10 Maintain Cases (Cont.)                     |
AP1200  11 Maintain Cases (Cont.)                     |
AC1209ST 12 Maintain Cases (Cont.)                   |
XHFILE  13 Maintain Cases (Cont.)                     |
XHCTYP  14 Maintain Cases (Cont.)                     |
AC1216  15 Maintain Cases (Cont.)                     |
AC1202  16 Maintain Cases (Cont.)                     |
AC1215  17 Maintain Cases (Cont.)                     |
AH1003  18 Maintain Cases (Cont.)                     |
AH1200  19 Maintain Cases (Cont.)                     |
XNALLOC 20 Maintain Cases (Cont.)                     |
XNPRCASE 21 Maintain Cases (Cont.)                   |
XNFFREE  22 Maintain Cases (Cont.)                   |
XNHCFLUP 23 Maintain Cases (Cont.)                   |
AH1200M1 24 Maintain Cases (Cont.)                   |
XNHLP50 25 Maintain Cases (Cont.)                   |
AC1214  26 Maintain Cases (Cont.)                   |
AC1212  27 Maintain Cases (Cont.)                   |
AC1207  28 Maintain Cases (Cont.)                   |
AC1205M3 29 Maintain Cases (Cont.)                   |
AH1000  30 Maintain Hearings                         |
AP1000  31 Maintain Parties                           |
AM1000  32 Maintain Members                           |
TM1000  33 Update Tables                               |
AC1203  34 Maintain File Locations                   |
FR1000  35 File Resubmit Report                       |
FR2000  36 File Location Report                       |
SE1000  37 Update Final Decisions                     |
CR1000  38 Decision Details Correction                 |
XNHLP  39 Decision Details Correction (Cont.)         |
WR1000  40 Written Reasons Audit Report               |
EC1000  41 View Cases                                 |
EC1000M1 42 View Cases (Cont.)                       |
EC1001  43 View Cases (Cont.)                       |
EC1002  44 View Cases (Cont.)                       |
EC1201  45 View Cases (Cont.)                       |
EC1212  46 View Cases (Cont.)                       |
EC1203  47 View Cases (Cont.)                       |
EC1216  48 View Cases (Cont.)                       |
EC1215M1 49 View Cases (Cont.)                      |
EC1207  50 View Cases (Cont.)                       |
-----

```


2001-02-18 13:11:44										**** CHART V3.1F for Natural ****										Page	56
User: HMC01										Index of Objects										Page	56
Object	Lngh	Library	Page	RC	Last Saved	Object	Lngh	Library	Page	RC	Last Saved	Object	Lngh	Library	Page	RC	Last Saved	Page	Lib: DEVLIB		
AC1000	700	DEVLIB	1	aG	1999-02-12	AC1204	790	DEVLIB	38	cO	1999-01-21	AC1206	240	DEVLIB	10	kH	1998-12-23	10	kH		
		DEVLIB	1	bD				DEVLIB	38	fE		AC1206M1					1	kL	<NoSource>		
		DEVLIB	1	bI				DEVLIB	38	hO		AC1206M1	100	DEVLIB	2	jF	1998-11-16	2	jF		
		DEVLIB	1	bJ		AC1204M1	290	DEVLIB	38	hJ	1998-12-23			DEVLIB	4	jC		4	jC		
		DEVLIB	1	eC		AC1204M2	230	DEVLIB	38	hH				DEVLIB	10	kE		10	kE		
		DEVLIB	2	eL		AC1205	1370	DEVLIB	1	iJ	1999-02-08	AC1207	390	DEVLIB	1	iA	1998-11-17	1	iA		
		DEVLIB	4	eM				DEVLIB	1	jK				DEVLIB	2	iM		2	iM		
		DEVLIB	10	gN				DEVLIB	2	iB				DEVLIB	4	hA		4	hA		
AC1000H1			9	bG	<NoSource>			DEVLIB	2	kK				DEVLIB	10	jL		10	jL		
			42	bG				DEVLIB	4	bG				DEVLIB	28	aG		28	aG		
AC1000M1	160	DEVLIB	1	bN	1998-10-27			DEVLIB	4	dL				DEVLIB	28	bH		28	bH		
		DEVLIB	9	aG				DEVLIB	4	hD		AC1207M1	120	DEVLIB	28	bF	1998-11-16	28	bF		
AC1001	290	DEVLIB	1	bB	1998-12-20			DEVLIB	10	JA		AC1207M2	110	DEVLIB	28	bG	1998-10-27	28	bG		
		DEVLIB	3	aG				DEVLIB	10	mK		AC1209	190	DEVLIB	1	cA	1998-11-17	1	cA		
		DEVLIB	3	bG				DEVLIB	38	hM				DEVLIB	1	jE		1	jE		
		DEVLIB	3	bH				DEVLIB	38	JN				DEVLIB	2	eE		2	eE		
AC1001H1			3	cF	<NoSource>	AC1205M1	80	DEVLIB	1	JC	1998-10-27			DEVLIB	2	kE		2	kE		
AC1001M1	110	DEVLIB	3	bF	1998-10-27			DEVLIB	2	kC				DEVLIB	4	dD		4	dD		
AC1002	290	DEVLIB	1	bA	1998-11-17			DEVLIB	4	dH				DEVLIB	4	eG		4	eG		
		DEVLIB	2	aG				DEVLIB	6	aG				DEVLIB	10	aG		10	aG		
		DEVLIB	2	bH				DEVLIB	10	mC				DEVLIB	10	bI		10	bI		
		DEVLIB	2	bI				DEVLIB	38	JO				DEVLIB	10	gI		10	gI		
		DEVLIB	3	bE		AC1205M2	150	DEVLIB	1	JJ	1999-02-08			DEVLIB	10	mE		10	mE		
AC1002H1			2	cG	<NoSource>			DEVLIB	2	kH				DEVLIB	38	JG		38	JG		
AC1002M1	150	DEVLIB	2	bG	1998-10-27			DEVLIB	4	dI		AC1209M1	100	DEVLIB	10	bE	1998-10-27	10	bE		
AC1201	430	DEVLIB	1	eN	1999-02-08			DEVLIB	10	mH		AC1209ST	200	DEVLIB	1	cC	1998-12-18	1	cC		
		DEVLIB	1	iL				DEVLIB	38	JJ				DEVLIB	1	JD		1	JD		
		DEVLIB	2	gH		AC1205M3	190	DEVLIB	1	JG	1998-10-27			DEVLIB	2	eB		2	eB		
		DEVLIB	2	iJ				DEVLIB	2	KI				DEVLIB	2	kD		2	kD		
		DEVLIB	4	gF				DEVLIB	4	dJ				DEVLIB	4	dC		4	dC		
		DEVLIB	4	hC				DEVLIB	10	mJ				DEVLIB	4	eE		4	eE		
		DEVLIB	10	iJ				DEVLIB	29	aG				DEVLIB	10	gH		10	gH		
		DEVLIB	10	JI				DEVLIB	38	JL				DEVLIB	10	mD		10	mD		
AC1202	730	DEVLIB	1	eH	1998-11-20	AC1205M4	180	DEVLIB	1	JI	1998-11-18			DEVLIB	12	aG		12	aG		
		DEVLIB	2	GI				DEVLIB	2	KA				DEVLIB	12	bH		12	bH		
		DEVLIB	4	GH				DEVLIB	4	dG				DEVLIB	38	JF		38	JF		
		DEVLIB	10	iG				DEVLIB	5	aG		AC1209S1	90	DEVLIB	12	bF	1998-11-05	12	bF		
		DEVLIB	16	aG				DEVLIB	10	mI		AC1210	1370	DEVLIB	10	bo	1999-02-10	10	bo		
		DEVLIB	16	bI				DEVLIB	38	JK				DEVLIB	10	do		10	do		
AC1202M1	150	DEVLIB	16	bE	1998-10-27	AC1205M5	90	DEVLIB	1	JF	1998-10-27	AC1210B	590	DEVLIB	10	dG	1998-11-17	10	dG		
AC1203	850	DEVLIB	1	eM	1999-02-08			DEVLIB	2	kF				DEVLIB	10	eG		10	eG		
		DEVLIB	2	gJ				DEVLIB	4	dE		AC1212	100	DEVLIB	1	gO		1	gO		
		DEVLIB	4	GI				DEVLIB	10	mF				DEVLIB	2	gL		2	gL		
		DEVLIB	10	iH				DEVLIB	38	JH				DEVLIB	4	gK		4	gK		
		DEVLIB	34	aG		AC1206	240	DEVLIB	1	iM	1998-12-23			DEVLIB	10	iK		10	iK		
		DEVLIB	34	bJ				DEVLIB	2	II				DEVLIB	27	aG		27	aG		
		DEVLIB	34	bK				DEVLIB	2	JI		AC1212M1	130	DEVLIB	27	bG	1998-10-27	27	bG		
AC1203M1	240	DEVLIB	34	bH	1998-11-06			DEVLIB	4	IB		AC1213	890	DEVLIB	38	CA	1998-12-24	38	CA		
AC1203M2	60	DEVLIB	34	bC	1998-10-27			DEVLIB	4	JF				DEVLIB	38	FI		38	FI		

2001-02-18 13:11:44		**** CHART V3.1F for Natural ****										Page 63					
User: HMC01		Index of Objects										Lib: DEVLIB					
Object	Lngh	Library	Page	RC	Last Saved	Object	Lngh	Library	Page	RC	Last Saved	Object	Lngh	Library	Page	RC	Last Saved
XNPARTYJ	140	DEVLIB	44	cL	1998-11-27	XNTAXYR	40	DEVLIB	2	eJ	1998-11-30						
XNPARTYT	200	DEVLIB	10	bH				DEVLIB	4	bF							
		DEVLIB	10	cH				DEVLIB	4	eL							
		DEVLIB	12	bE				DEVLIB	10	gM							
		DEVLIB	12	cE		XNTITLE	50	DEVLIB	10	dL							
XNPARTY2	120	DEVLIB	18	lA				DEVLIB	10	eE							
		DEVLIB	19	fC				DEVLIB	18	lE							
XNPRCASE	1540	DEVLIB	1	eG	1999-01-27			DEVLIB	19	fG							
		DEVLIB	1	gJ		XNYEAR	50	DEVLIB	1	bG	1998-11-25						
		DEVLIB	2	eD				DEVLIB	1	cB							
		DEVLIB	2	hG				DEVLIB	2	eH							
		DEVLIB	4	gN				DEVLIB	4	bC							
		DEVLIB	4	iO				DEVLIB	4	eH							
		DEVLIB	10	iN				DEVLIB	10	gK							
		DEVLIB	10	kO				DEVLIB	30	bE							
		DEVLIB	18	bL				DEVLIB	37	bE							
		DEVLIB	18	dN				DEVLIB	38	bH							
		DEVLIB	21	aG				DEVLIB	40	bG							
		DEVLIB	21	bI				DEVLIB	41	bH							
		DEVLIB	41	eK				DEVLIB	53	bG							
		DEVLIB	41	gI													
		DEVLIB	44	gL													
		DEVLIB	44	hI													
		DEVLIB	53	dO													
XNPRTER	70	DEVLIB	1	eK	1998-11-30												
		DEVLIB	2	gO													
		DEVLIB	4	gG													
		DEVLIB	10	iO													
		DEVLIB	18	bE													
		DEVLIB	18	dG													
		DEVLIB	33	bC													
		DEVLIB	33	dN													
		DEVLIB	33	fG													
		DEVLIB	41	eF													
		DEVLIB	41	gE													
		DEVLIB	44	eH													
		DEVLIB	44	hF													
		DEVLIB	53	dJ													
XNSTATUS	230	DEVLIB	1	cH	1999-02-08												
		DEVLIB	1	gF													
		DEVLIB	2	cK													
		DEVLIB	2	hC													
		DEVLIB	4	cN													
		DEVLIB	4	iK													
		DEVLIB	21	bD													
		DEVLIB	41	cF													
		DEVLIB	44	cK													
XNTAXYR	40	DEVLIB	1	eE	1998-11-30												

HMC0045: Total of 220 different objects drawn (excl. don't explodes) with a total of approx. 50360 lines of source code.

Object	Object	Object	Object	Object	Object	Object	Object	Object	Object	Object	Object
2001-02-18 13:11:50	**** CHART V3.1F for Natural ****										Page 64
User: HMC01	Objects NOT Referenced										Lib: DEVLIB
AA1000	AS1001	FC1000M1	HEXLDTRN	LC1100H2	NH1000	TA1300	VB2000				
AA1000B	AS1001B	FR1000B	HEXLICEN	LC1100M1	NH1000B	TA1300M1	VB2000M1				
AA1000M1	CC1000M1	FR2000B	HEXTRNLD	LC1100M2	NH1000M1	TC1000	WR1000B				
AC1205AA	CF1000	GLOBAREA	HP1000	LC1100M3	NL1000	TC1000B	XF1000				
AC1205MA	CF1000M1	HD1000	HP1000M1	LC2000	NL1000M1	TC1000M1	XHAGCY				
AC1205OR	CF1001B	HD1000M1	HP1100	LC2000H1	NL2000B	TD1004	XHAGCY1				
AC1210ST	CH	HD1001B	HP1100M1	LC2000M1	NS	TD1005	XHBOOK				
AC2200AA	CHART	HD1002B	HP1200	LC2100	PL1000	TIM	XHCGRPS				
AHLISTST	CM1000	HD1003B	HP1200H1	LC2100M1	PL1000M1	TIM-AUTH	XHCTVPS				
AH1100M2	CM1000M1	HD2000	HP1200M1	LC3000	PL2000	TIM-CASE	XHDTVPS				
AF1400	CM1001B	HD2000M1	HR1000	LC3000M1	PRNTCASE	TIM-DEL	XHGGRP				
AF1400B	CM1002B	HD2001B	HR1000M1	LC3100B	RB1000	TIM-DUPL	XHGGRP				
AF1400M1	CM2000	HE1000	HR1001B	LC4000	RB1000M1	TIM-LCID	XHLOCN				
AF2000	CM2000M1	HE1000B	HS1004	LC4000M1	RB2000	TIM-LOOK	XHPARTY				
AF2000M1	CM2001B	HE1000M1	HS1004B	LC4000M2	RB2000M1	TIM-RSB2	XHREPT				
AF2002B	CM2002B	HE1001	HS1004M1	LOAD	RB2000M2	TIM-RSUB	XHSOLCS				
AF3000	CNTCASEB	HE1001B	HU1000	LOADAG	RB2000N	TIMBATCH	XHSTUS				
AF3000M1	CNTCASES	HE1001M1	HU1000B	LOADCIRC	RD1000	TIMONLIN	XHTABL				
AF3002B	CS1000	HE1002	HU1000C	LOADCS	RD1000B	TR1000	XHTRIB				
AF4000	CS1000M1	HE1002B	HU1000M1	LOADCS2	RD1000M1	TR1000B	XNCALC				
AF4000B	CS1001B	HE1002M1	ID1000	LOADHR	REMAAT	TR1000M1	XNCALC2				
AF4000M1	CS1002B	HE1003	ID1000B	LOADLC	REMAAT2	TR2000	XNCALC3				
ARXLAS2	DC1000	HE1003B	ID1000M1	LOADMEMB	SC1000	TR2000B	XNCONF				
AR6101	DC1000B	HE1003M1	IR1000	LOADMEMB	SC1000B	TR2000M1	XNCONFM1				
AR6101C	DC1000M1	HEX#DATE	IR1000B	LOADM1	SC1000M1	TV1000	XNGHELP				
AR6101M1	DC2000	HEX#DISP	IR1000C	LOADPTS	SD1000	TV1000M1	XNHEAR				
AR6103Y	DC2000M1	HEX#DONT	IR1000D	LOAD1	SD1000B	TV1001	XNHEARM1				
AR6103YM	DC2001	HEX#DRAW	IR1000E	LOAD1H1	SD1000C	TV1001M1	XNHELPM3				
AR6103Y1	DE1000	HEX#EXC	IR1000F	LOAD1M1	SD1000M1	TV1001M2	XNHLP1				
AR6103Y2	DE1000B	HEX#EXCM	IR1000G	LOAD2	SR100VM2	TV1002	XNINCR				
AR6107C	DE1000C	HEX#EXPR	IR1000H	LOAD2M1	SR100V2	TV1002M1	XNINCRS				
AR6107C1	DE1000D	HEX#INIT	IR1000I	MB1000	SR100VM2	TV1002M2	XNINCR4				
AR6107MC	DE1000M1	HEX#LANG	IR1000J	MB1000M1	SR101V2	USR-FLD	XNINCR5				
AR6109	DE2000B	HEX#LCFM	IR1000K	MB1001	SR101V2M	USR-MSG	XNLINK				
AR6109C	DE2000C	HEX#LCMM	IR1000MS	MB2000	SR1100B	USR0070P	XNLINKM1				
AR6109M1	DE2000D	HEX#LINE	IR1000M1	MB2000M1	SR1100H	USR0070T	XNLIST				
AR6111Y	DOWNLDE	HEX#LNGI	IR1000SA	MB2001	SR1200B	USR0330P	XNPARTYN				
AR6111YM	DOWNLM1	HEX#LNGO	IR1000SB	MB2001M1	SR2000	USR0330T	XNPARTYV				
AR6111Y1	DOWNLOAD	HEX#LNUM	IR1000SC	MB3000	SR2000M1	USR1025P	XNPRCAST				
AR6111Y2	EC1204	HEX#MSG1	IR1000SD	MB3001	SR2100B	USR1025T	XNTEST				
AR6200	EC1204M1	HEX#MTHD	IR1000SE	MB3001M1	SR3000M1	USR1029P	XNYEAR				
AR6200M1	EC1207AA	HEX#OBJT	IR1000SF	MB4000	SYSINFOP	USR1029T	XXGLOBS				
AR6201C	EC1213	HEX#ORD	IR1000SG	MB4000M1	SYSINFOP	USR1032P	Y2KCONV				
AR6301M1	EC1213M1	HEX#ORD1	IR1000SH	MB4001	SYSINFOP	USR1032T	Y2KCONV1				
AR7200	EC1213M2	HEX#STP	IR1000SI	MC1000	SYSINFOT	USR1035P	Y2KCONV1				
AR7200M1	EP1100	HEX#STFM	IR1000SJ	MC1000B	TA1000	USR1035T	Y2KCONV2				
AR7210	EP1100M1	HEX#SUBR	IR1000ST	MC1000M1	TA1100	USR1057P	Y2KPHOL				
AR7220	EP1200AA	HEX#WFD1	LC1000	MI1000	TA1100M1	USR1057T	ZJOB1				
AS1000	FC1000	HEX#WFD2	LC1000M1	MI1000B	TA1200	VB1000	ZMAINGDA				
AS1000M1	FC1000B	HEX#WFPD	LC1100	MI1000M1	TA1200M1	VB1000M1					

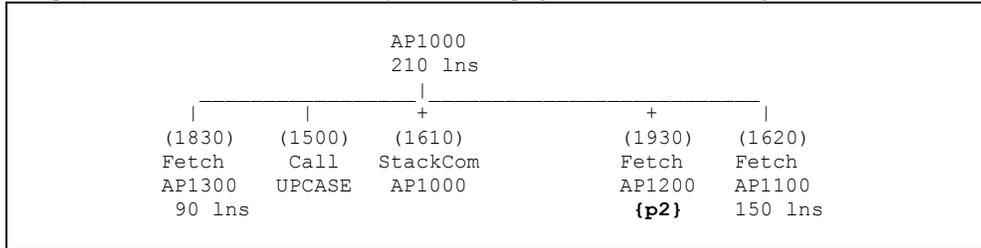
10.6 Less redundancy with special Don't explode values

CMWKF01, CMWKF02, and CMWKF04 are three special **Don't explode** values that can lessen redundant *CHART*-ing and improve clarity.

For example, if you use 'CMWKF01' as one of your "don't explodes" then all seeds in the 'seed list' (ie. in file CMWKF01) will be treated as "don't explodes" within every 'chart':

```
LOGON DEVLIB
CHART CMWKF01, GO, 0, CMWKF01, MENU*
FIN
```

An object that is also a 'seed' in CMWKF01 will still be drawn, but with a '+' to indicate it has NOT been exploded. The page where the object/seed is fully exploded will also be shown below the object name. For example, "{p2}" means that the seed is fully drawn on page 2 of the current high-volume CHART-ing job:



If you use 'CMWKF02' as one of your "don't explodes" then all the seeds in the 'original seeds plus extra seeds' (ie. in file CMWKF02) will be treated as "don't explodes" within every 'chart':

```

LOGON DEVLIB
CHART CMWKF01, GO, 0, CMWKF02, MENU*
FIN
  
```

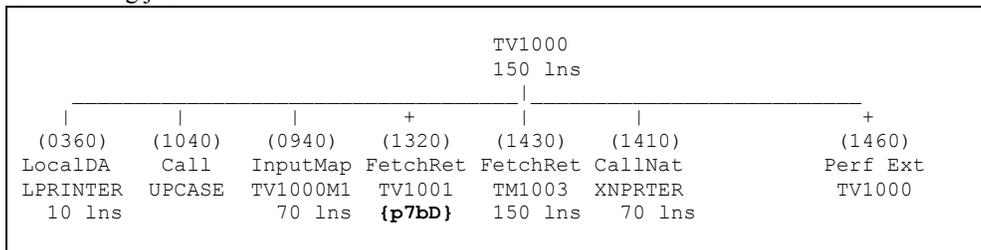
An object that is also a 'seed' in CMWKF01 will still be drawn, but with a '+' to indicate it has NOT been exploded. The page where the object/seed is fully exploded will also be shown below the object name.

If you use 'CMWKF04' as one of your "don't explodes" then all the objects in the 'details of all objects chart-ed in CSV format' (ie. in file CMWKF04) will be treated as "don't explodes" within every (subsequent) 'chart'. This option stops redundant CHART-ing because each object with children will only be "exploded" once:

```

LOGON DEVLIB
CHART CMWKF01, GO, 0, CMWKF04, MENU*
FIN
  
```

Such an object will still be drawn, but with a '+' to indicate it has NOT been exploded. The page, row, and column where the object was originally exploded will also be shown below the object name. For example, "{p7bD}" means that the object was originally drawn at row 'b', column 'D' on page 7 of the current high-volume CHART-ing job:



Note: If CMWKF04 is used as special "don't explode" value, then you must also define CMWKF05 in your JCL, with the same record format and record length as CMWKF04.

10.7 Example CHARTs using special Don't explode values

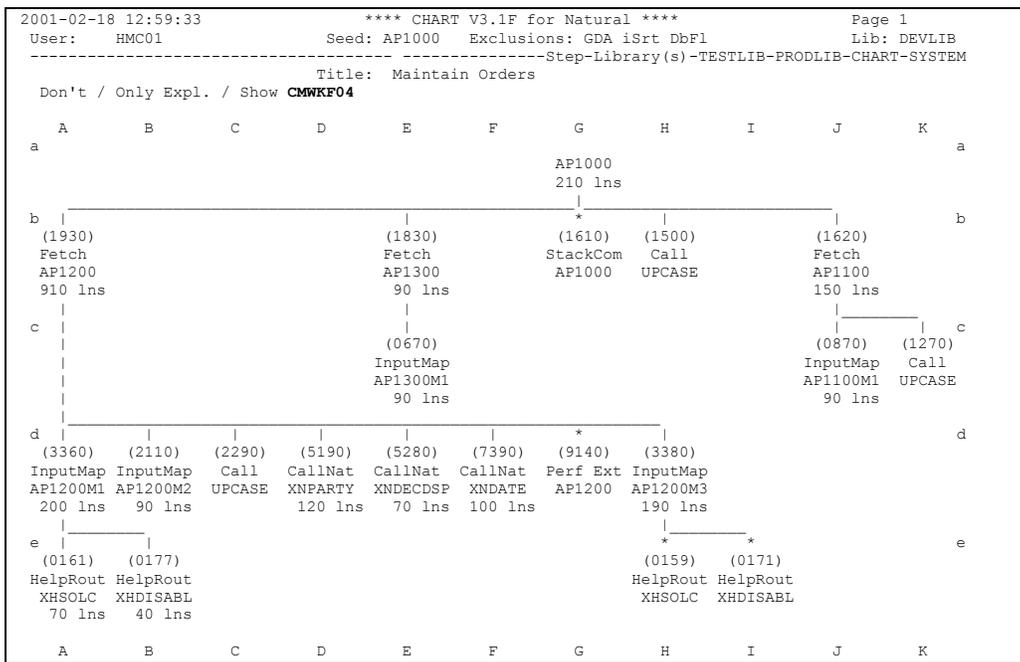
Simple examples are shown on the following pages to illustrate the effect these special “don’t explode” values have. The examples all use a ‘seed list’ (ie. CMWKF01) containing two seeds (AP1000 and AP1200) as follows:

```
BROWSE      HMC01.ORIGINAL.SEEDS                Line 00000000 Col 001 080
  Command ==>                                Scroll ==> CSR
***** Top of Data *****
AP1000      Maintain Orders
AP1200      View Orders
***** Bottom of Data *****
```


The third example was generated using the *CHART* command:

```
LOGON DEVLIB
CHART CMWKF01,GO,99,CMWKF04
FIN
```

with a special “don’t explode” value of ‘CMWKF04’ used. Two pages resulted:



```

2001-02-18 12:59:41          **** CHART V3.1F for Natural ****          Page 2
User:      HMC01              Seed: AP1200  Exclusions: GDA iSrt DbFl      Lib: DEVLIB
-----Step-Library(s)-TESTLIB-PRODLIB-CHART-SYSTEM
Title:      View Orders
Don't / Only Expl. / Show CMWKF04

  A      B      C      D      E      F      G      H      I      J      K      a
a
                                     AP1200
                                     910 lns
b
+-----+-----+-----+-----+-----+-----+-----+-----+
(3360)  (2110)  (2290)  (5190)  (5280)  (7390)  (9140)  (3380)
InputMap InputMap Call  CallNat CallNat CallNat Perf Ext InputMap
AP1200M1 AP1200M2 UPCASE XNPARTY XNDEC DSP XNDATE AP1200 AP1200M3
{p1dA}  90 lns      120 lns  70 lns  100 lns      {p1dH}
  A      B      C      D      E      F      G      H      I      J      K

```

AP1200M1 and AP1200M3 are drawn (and exploded) on page 1 at 'dA' and 'dH' respectively. At this time, they are also added to CMWKF04 (the work file holding details of all objects CHART-ed). When AP1200 (and its child objects) are drawn on page 2, AP1200M1 and AP1200M3 are NOT 'exploded' again because they were found in CMWKF04. The special "don't explode" value of 'CMWKF04' has removed some redundancy from the CHARTs.

10.8 Changing excluded object types and Don't explode objects

Object types to be **excluded** and **Don't explode** objects can be changed during high-volume *CHART*-ing (or even before the first 'seed' is drawn). This is achieved with a special command embedded in the seed 'dataset'. The charts generated for your application can be tailored in this way for maximum impact and coverage.

For example, it may be useful to **exclude** more object types from a particular chart to improve its clarity. Also, a chart too large for one page (and therefore with many **unplaced** objects) could be split into two charts:

- specify a significant 'parent' object from the chart as a **Don't Explode** object,
- this significant 'parent' object would have its 'child' objects suppressed,
- this object could then be included in the seed 'dataset' and charted separately.

To change the **excluded** / **Don't explode** criteria during high-volume *CHART*-ing:

- locate the seed in the seed 'dataset' to which the new criteria are to apply,
- insert a new line in the seed 'dataset' immediately before that seed,
- enter '+++++++' in the first eight characters of this line, followed by a space (this indicates that new **exclusion** and **Don't explode** criteria are being defined),
- then enter your new **excluded** object types in the next **ten** characters, followed by a space,
- then enter your new **Don't explode** objects, separated by spaces or commas.

If all the **Don't explode** objects won't fit into this line, repeat the '+++++++' line and enter the remaining **Don't explode** objects. The '+++++++' and **excluded** object types should be left as in the original line. Note: The new **exclude** / **Don't explode** criteria will remain in effect for all subsequent seeds, or until new criteria are defined.

Example of new criteria defined during high-volume *CHART*-ing - the new criteria (**exclude** Map, DataArea, and Copycode, **Don't explode** ERRORHDL, MENUHDL, CHKRSTRT, COM100N1) apply to seed IMP200P1 and subsequent seeds:

```
.  
.   
IMP100P1 Staff Impact Report Submission  
+++++++ MDC          ERRORHDL MENUHDL CHKRSTRT COM100N1  
IMP200P1 Staff History Report  
.   
.
```

10.9 Changing the Natural Library and Step Libraries

The current **Natural Library** and **Step Libraries** can be changed during high-volume *CHART*-ing (or even before the first 'seed' is drawn). This is achieved with a special command embedded in the seed 'dataset'. In this way, charts for your whole application can be generated no matter which **Natural Library** the seeds are held on.

To change the **Natural Library** during high-volume *CHART*-ing:

- locate the seed in the seed 'dataset' to which the new **Natural Library** is to apply,
- insert a new line in the seed 'dataset' immediately before that seed,
- enter '>>>>>>>>' in the first eight characters of this line, followed by a space (this indicates that a new **Natural Library** is being defined),
- then enter the new **Natural Library** in the next eight characters,
- then enter any specific **Step Libraries**, separated by spaces / commas. Otherwise, the **Step Libraries** will default to those defined in the initial 'CHART' program.

Please note that the new **Natural Library** and **Step Libraries** will remain in effect for all subsequent seeds, or until new **Libraries** are defined.

Example of a new **Natural Library** and three new **Step Libraries** defined during high-volume *CHART*-ing - the new **Natural Library** will apply to seed IMP200P1 and subsequent seeds:

```
.  
.   
IMP100P1 Staff Impact Report Submission  
>>>>>>>> HISTLIB PROD TEST DEV  
IMP200P1 Staff History Report  
.   
.
```

10.10 Potential NAT1205 (No more sort work space) Error

This error may occur when the **index of objects** is being generated. That is, all the *CHART*s have been generated, the **table of contents** has been generated, and the **index of objects** is about to be generated.

To generate the **index of objects**, CMWKF03 needs to be sorted into 'object name' x 'library' order. CMWKF03 holds one record for every object drawn in every *CHART* for the current batch job. If you are generating a large number of *CHART*s in one batch job, CMWKF03 can become very large. When the *CHART* software comes to sort this file before generating the **index of objects**, it is possible a NAT1205 (No more sort space) error will occur. That is, there is a limit to how much you can sort 'in core' with Natural.

If you do receive a NAT1205, the **index of objects** (and the **not referenced** list, if required) can be generated as a separate job after the *CHART*s and **table of contents** have been generated.

To do this, you must first sort the dataset referred to in CMWKF04. This dataset should be sorted into ascending order using (at least) columns 1 to 67.

If you are using JCL to sort the dataset, the SORT command would be:

```
SORT FIELDS=(1,66,CH,A)
```

If you are using ISPF-Edit to sort the dataset, the command would be:

```
SORT 1 66 A
```

After the dataset is sorted, you can generate the **index of objects** and **not referenced list** in a separate batch job. An example of the JCL to run this job is shown on the next page. (Note: If the SORT and 'generate' are in the same job, the SORT step must be run before the 'generate'.)

A sample of the JCL required to generate the **index of objects** (and **not referenced list**, if required) follows:

```
//<userid>C JOB 'CHART V2',CLASS=A,MSGCLASS=X,NOTIFY=<userid>
//*
//NATBATCH EXEC PGM=...
//STEPLIB DD DSN=...
// DD DSN=...
//DDCARD DD DSN=...
//*
//CMPRINT DD SYSOUT=*
//CMPRT01 DD SYSOUT=*
//CMPRT02 DD SYSOUT=*
//CMPRT03 DD DSN=<userid>.INDEX.OF.OBJECTS,
// DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(2,2),RLSE),
// UNIT=SYSDA,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=13300)
//CMPRT04 DD DSN=<userid>.SURPLUS.OBJECTS,
// DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(2,2),RLSE),
// UNIT=SYSDA,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=13300)
//*
//CMWKF03 DD DSN=<userid>.DATA.FOR.INDEX.OF.OBJECTS,
// DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(5,5),RLSE),
// UNIT=SYSDA,DCB=(RECFM=FB,LRECL=20,BLKSIZE=16000)
//CMWKF04 DD DSN=<userid>.CHARTED.OBJECTS.CSV.FORMAT,
// DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(10,10),RLSE),
// UNIT=SYSDA,DCB=(RECFM=FB,LRECL=160,BLKSIZE=16800)
//*
//CMSYNIN DD *
LOGON <library>
GLOBALS IM=D
CHART CMWKF04,INDEX|NOTREF|BOTH[,<last-page-num>]
FIN
/*
//
```

CHART CMWKF04,INDEX|NOTREF|BOTH[,<last-page-num>]

If the option 'INDEX' is used, only the **index of objects** will be generated. If the option 'NOTREF' is used, only the **not referenced list** will be generated. If the option 'BOTH' is used, the **index of objects** and the **not referenced list** will be generated.

The parameter <last-page-num> is optional. If supplied, the page numbering for the generated report(s) will begin after this number. For example, if the 'INDEX' option is used with a <last-page-num> of 37, then the generated **index of objects** would have a page number of 38 on its first page.

Blocksize for these datasets can be any multiple of the record length and should conform to your site's record blocking conventions. Do NOT use a dataset DISPosition of MOD (for MODify) for work files in your JCL as this will lead to unpredictable results during the generation of the "Index of Objects" and "Table of Contents".

10.11 Details of CHART-ed objects held in CSV Format on CMWKF04

Every object drawn in any *CHART* (in batch) has information about it recorded in CMWKF04. This information is recorded in CSV (ie. Comma Separated Value) format suitable for loading as a spreadsheet or as data for an analysis package. The fields (and their lengths) recorded for each object are:

Object-name	30
Library	8
Database	8
Date-saved	10
Page-number/Row/Column	6
Object-length	5
Number-of-children	3
Children-placed-flag	1
Calling-method	8
Calling-source-code-line-number	6
Parent-object-name	29
Parent-Page/Row/Column	6
Current-user-id	8
Current-date	10
Current-time	8

The following is an example of the information written to CMWKF04:

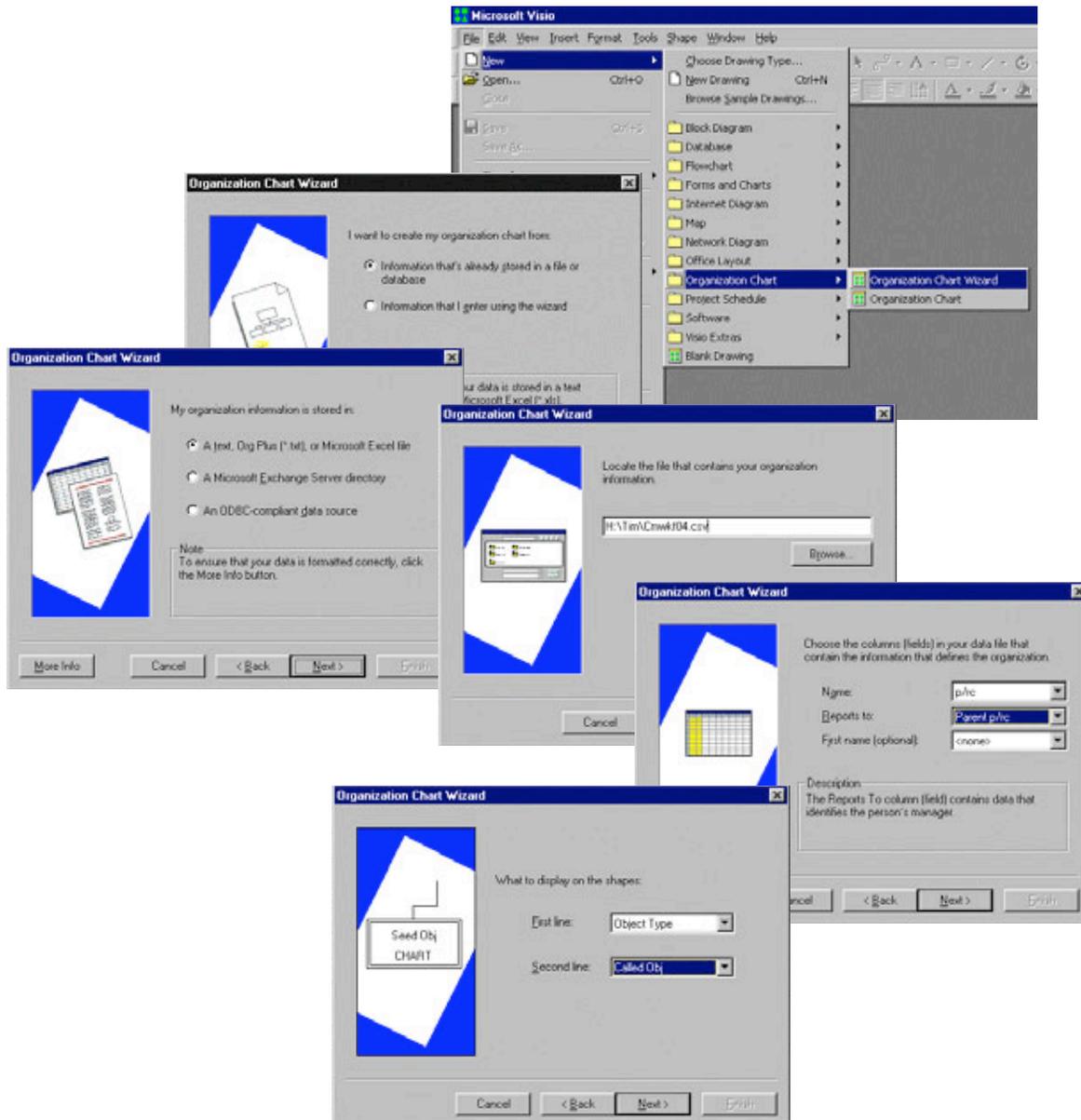
```

Called Obj.Lib.Dbase.Saved.p/rc.Length.Child.Child Placed?.Object Type,Called from Line.Parent Obj.Parent p/rc,Drawn by.Date Drawn,Time Drawn
HMC#WFD1,CHART,1999-03-25,0001aG,0060,001,Y,Seed Obj,(0000),0000,HMC01,2001-04-02,08:36:4
File1,CHART,<NoSource>,0001bG,9999?,000,Y,ReadWork,(0240),HMC#WFD1,0001aG,HMC01,2001-04-02,08:36:4
HMC#WFD2,CHART,1999-03-25,0002aG,0060,001,Y,Seed Obj,(0000),0000,HMC01,2001-04-02,08:36:5
File1,CHART,<NoSource>,0002bG,9999?,000,Y,ReadWork,(0300),HMC#WFD2,0002aG,HMC01,2001-04-02,08:36:5
HMC#WFD3,CHART,1999-03-25,0003aG,0090,001,Y,Seed Obj,(0000),0000,HMC01,2001-04-02,08:36:5
File3,CHART,<NoSource>,0003bG,9999?,000,Y,ReadWork,(0320),HMC#WFD3,0003aG,HMC01,2001-04-02,08:36:5
HMC#WFH4,CHART,2001-03-29,0004aG,0020,001,Y,Write Obj,(0000),0000,HMC01,2001-04-02,08:36:5
File4,CHART,<NoSource>,0004bG,9999?,000,Y,WriteWork,(0140),HMC#WFH4,0004aG,HMC01,2001-04-02,08:36:5
HMC#WFPG,CHART,1999-03-25,0005aG,0040,004,Y,Seed Obj,(0000),0000,HMC01,2001-04-02,08:36:5
File1,CHART,<NoSource>,0005bE,9999?,000,Y,ReadWork,(0150),HMC#WFPG,0005aG,HMC01,2001-04-02,08:36:5
File2,CHART,<NoSource>,0005bF,9999?,000,Y,WriteWork,(0310),HMC#WFPG,0005aG,HMC01,2001-04-02,08:36:5
File2,CHART,<NoSource>,0005bG,9999?,000,Y,ReadWork,(0340),HMC#WFPG,0005aG,HMC01,2001-04-02,08:36:5
HMC#WF01,CHART,1999-03-25,0005bH,0010,001,Y,FetchRet,(0350),HMC#WFPG,0005aG,HMC01,2001-04-02,08:36:5
File1,CHART,<NoSource>,0005cH,9999?,000,Y,WriteWork,(0120),HMC#WF01,0005bH,HMC01,2001-04-02,08:36:5
HMC#WFUP,CHART,1999-03-25,0006aG,0120,004,Y,Seed Obj,(0000),0000,HMC01,2001-04-02,08:36:5
File1,CHART,<NoSource>,0006bE,9999?,000,Y,ReadWork,(0440),HMC#WFUP,0006aG,HMC01,2001-04-02,08:36:5
File2,CHART,<NoSource>,0006bF,9999?,000,Y,WriteWork,(0500),HMC#WFUP,0006aG,HMC01,2001-04-02,08:36:5
File2,CHART,<NoSource>,0006bG,9999?,000,Y,ReadWork,(0560),HMC#WFUP,0006aG,HMC01,2001-04-02,08:36:5
HMC#WF01,CHART,1999-03-25,0006bH,0010,001,Y,FetchRet,(0760),HMC#WFUP,0006aG,HMC01,2001-04-02,08:36:5
File1,CHART,<NoSource>,0006cH,9999?,000,Y,WriteWork,(0120),HMC#WF01,0006bH,HMC01,2001-04-02,08:36:5
HMC#WF01,CHART,1999-03-25,0007aG,0010,001,Y,Seed Obj,(0000),0000,HMC01,2001-04-02,08:36:5
File1,CHART,<NoSource>,0007bG,9999?,000,Y,WriteWork,(0120),HMC#WF01,0007aG,HMC01,2001-04-02,08:36:5
HMC#WF02,CHART,1999-03-25,0008aG,0010,001,Y,Seed Obj,(0000),0000,HMC01,2001-04-02,08:36:5
File2,CHART,<NoSource>,0008bG,9999?,000,Y,WriteWork,(0120),HMC#WF02,0008aG,HMC01,2001-04-02,08:36:5
HMC#WF03,CHART,1999-03-25,0009aG,0010,001,Y,Seed Obj,(0000),0000,HMC01,2001-04-02,08:37:0
File3,CHART,<NoSource>,0009bG,9999?,000,Y,WriteWork,(0110),HMC#WF03,0009aG,HMC01,2001-04-02,08:37:0
HMC#WF04,CHART,2001-03-29,0010aG,0060,001,Y,Seed Obj,(0000),0000,HMC01,2001-04-02,08:37:0
File4,CHART,<NoSource>,0010bG,9999?,000,Y,WriteWork,(0450),HMC#WF04,0010aG,HMC01,2001-04-02,08:37:0
HMC#WF12,CHART,1999-03-25,0011aG,0020,002,Y,Seed Obj,(0000),0000,HMC01,2001-04-02,08:37:0
File1,CHART,<NoSource>,0011bF,9999?,000,Y,ReadWork,(0110),HMC#WF12,0011aG,HMC01,2001-04-02,08:37:0
File2,CHART,<NoSource>,0011bG,9999?,000,Y,WriteWork,(0180),HMC#WF12,0011aG,HMC01,2001-04-02,08:37:0
HMC#WF43,CHART,1999-03-25,0012aG,0070,002,Y,Seed Obj,(0000),0000,HMC01,2001-04-02,08:37:0
File4,CHART,<NoSource>,0012bF,9999?,000,Y,ReadWork,(0500),HMC#WF43,0012aG,HMC01,2001-04-02,08:37:0
HMC#WF03,CHART,1999-03-25,0012bG,0010,001,Y,FetchRet,(0660),HMC#WF43,0012aG,HMC01,2001-04-02,08:37:0
File3,CHART,<NoSource>,0012cG,9999?,000,Y,WriteWork,(0110),HMC#WF03,0012bG,HMC01,2001-04-02,08:37:0
HMC#WF45,CHART,1999-03-25,0013aG,0020,002,Y,Seed Obj,(0000),0000,HMC01,2001-04-02,08:37:0
File4,CHART,<NoSource>,0013bF,9999?,000,Y,ReadWork,(0130),HMC#WF45,0013aG,HMC01,2001-04-02,08:37:0
File5,CHART,<NoSource>,0013bG,9999?,000,Y,WriteWork,(0150),HMC#WF45,0013aG,HMC01,2001-04-02,08:37:0
File5,CHART,<NoSource>,0014bF,9999?,000,Y,ReadWork,(0130),HMC#WF54,0014aG,HMC01,2001-04-02,08:37:0
File4,CHART,<NoSource>,0014bG,9999?,000,Y,WriteWork,(0150),HMC#WF54,0014aG,HMC01,2001-04-02,08:37:0
HMC#WILD,CHART,2001-03-28,0015aG,0140,006,Y,Seed Obj,(0000),0000,HMC01,2001-04-02,08:37:1
HMC#DATE,CHART,1999-03-25,0015bD,0070,000,Y,CallNat,(0670),HMC#WILD,0015aG,HMC01,2001-04-02,08:37:1
HMC#INFOF,CHART,1999-01-27,0015bE,0080,000,Y,CallNat,(0850),HMC#WILD,0015aG,HMC01,2001-04-02,08:37:1
HMC#INFON,CHART,1999-03-25,0015bF,0090,000,Y,CallNat,(0870),HMC#WILD,0015aG,HMC01,2001-04-02,08:37:1
HMC#MSG1,CHART,2001-03-26,0015bG,0430,000,Y,CallNat,(1360),HMC#WILD,0015aG,HMC01,2001-04-02,08:37:1
HMC#INFO,CHART,<NoSource>,0015bH,9999?,000,Y,CallNat#, (0890),HMC#WILD,0015aG,HMC01,2001-04-02,08:37:1
HMC#WF01,CHART,1999-03-25,0015bI,0010,001,Y,FetchRet,(1190),HMC#WILD,0015aG,HMC01,2001-04-02,08:37:1
File1,CHART,<NoSource>,0015cI,9999?,000,Y,WriteWork,(0120),HMC#WF01,0015bI,HMC01,2001-04-02,08:37:1

```

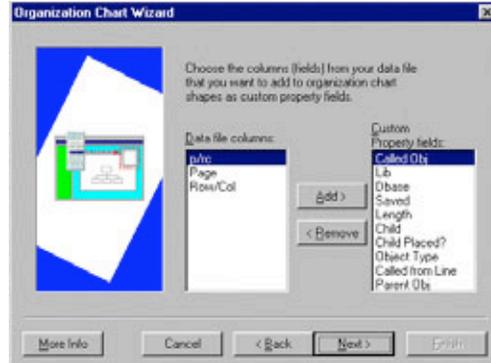
10.12 Importing CHART-ed objects into Visio2000

Every object drawn in any *CHART* (in batch) has information about it recorded in CMWKF04. This information can be imported directly into Microsoft Visio2000 as “organisation charts”. Download the dataset written to by NATURAL file CMWKF04 to the platform on which you use Visio2000. Follow these steps to import the resulting file into Visio2000. Note that on the fifth screen, the “Name” field **must** be “p/rc” (i.e. page row/column) and the “Reports to” field **must** be “Parent p/rc”:

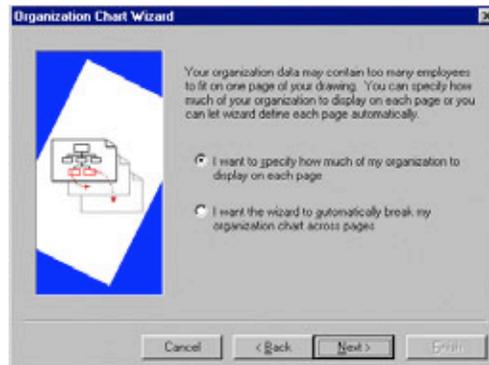


The “First line” field must be “Object Type” and the “Second line” field must be “Called Obj”:

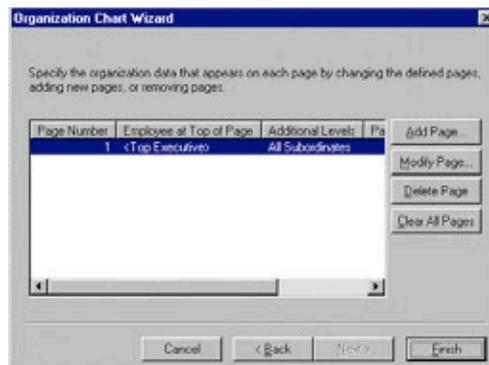
“Add” as many columns as you like to the list of “Custom Property fields”:



Choose the “I want to specify how much of my organization to display on each page” option and on the next screen leave the defined pages option as shown below:



Click the “Finish” button and Visio will import the data from your CMWKF04 data file as a set of “Organization Chart”’s.



11 Messages used in CHART

- HMC0001: No "SEED" Object Name supplied for Chart.
HMC0002: Object <object name> NOT found in Library or Step Libs.
HMC0003: Chart Exclusion: <exclusion> NOT valid.
HMC0004: Type "R" includes Types "S", "N", and "H".
HMC0005: Type "D" includes Types "G", "L", and "A".
HMC0006: Type "O" includes Types "I" and "F".
HMC0010: Choose a Valid PF Key or PF3 to Quit.
HMC0012: Trace Interval must be zero or greater.
HMC0013: Use only one "*" to indicate group of objects: <value>
HMC0014: "*" cannot be in this position all objects will be suppressed: <value>
HMC0015: "*" must be the last significant character: <value>
HMC0016: At least one significant character must be used with "!".
HMC0017: Invalid object type found after the ")": <value>
HMC0018: No object types were found after the ")": <value>
HMC0019: Your specified limit of Charts has been reached. Chart-ing terminated.
HMC0020: Limit of objects reached -- Chart for <seed name> may be incomplete.
HMC0021: Don't Show indicator can only be in first character of object: <value>
HMC0022: More than 20 "Don't Explodes". Ignoring <value>
HMC0023: "Don't Explode" over 12 characters. Ignoring <value>
HMC0024: More than 10 "Step Libraries". Ignoring <value>
HMC0025: "Step Library" over 8 characters. Ignoring <value>
HMC0026: Enter Object Name.
HMC0027: Object <object name> was NOT found in Chart.
HMC0028: Object <object name> found at row/column: <row/column>
HMC0029: Press PF12 to view Objects with Descendants NOT Drawn.
HMC0030: "Not Referenced" List is NOT Available yet for UNIX.
HMC0031: All Objects in this Library were Referenced.
HMC0033: Line Numbers not available. Chart is too large.
HMC0034: Connector can NOT be blank.
HMC0035: Must be: BLue, GReen, NNeutral, PInk, REd, TURquoise, or YELlow
HMC0036: Must be: "B"linking, "D"efault, "I"ntense, or Reverse "V"ideo.
HMC0037: At least one Repository must be selected.
HMC0038: Source Code Access subprograms for Repository(s) not found.
HMC0039: Children of Inline Subroutines NOT drawn at: <row/column> (and can't be drawn on later CHARTs).
HMC0040: Object name is longer than eight characters.
HMC0042: Date Order must be one of "DMY", "DYM", "MDY", "MYD", "YDM", or "YMD".
HMC0043: Answer "N" for NO separators in date, or "Y" and choose a separator.

HMC0044: Too many objects for "Index of Objects". Use CHART CMWKF04 in batch.
HMC0045: Total of NATURAL <count> objects drawn (excl. don't explodes) with a total of approx. <count> lines of source code.
HMC0046: <library name> will not fit into step-lib array. <library-name> will be used instead.
HMC0047: <option> is NOT valid. Use INDEX, NOTREF, or BOTH after seed of CMWKF04.
HMC0048: <object name> is less than <object name> - CMWKF04 must be SORTed by object name (chars 1-30) at least.
HMC0049: Natural Security is installed. Use USR1025N option instead of NONATSEC.

- HMC0050: <PF key> has no effect while the Chart is Zoom-ed.
HMC0051: Already at top of Chart. <PF key> ignored.
HMC0052: Already at bottom of Chart. <PF key> ignored.
HMC0053: Already at far left of Chart. <PF key> ignored.
HMC0054: Already at far right of Chart. <PF key> ignored.
HMC0055: Already showing left of object names. <PF key> ignored. Try PF11 (Right).
HMC0056: Already showing right of object names. <PF key> ignored. Try PF10 (Left).
HMC0057: Complete Chart already shown. No need to Zoom. <PF key> ignored.
HMC0058: Left side of Object Names shown. Use PF11 to see rest of names.
- HMC0059: Display: source line number from which object was called.
HMC0060: Display: length of called object (rounded to nearest 10).
HMC0061: Display: user who saved source code of called object.
HMC0062: Display: date source code was saved. PF21 to see century.
HMC0063: Display: library where source of called object was found.
HMC0064: Display: method in which object was called.
HMC0065: Display: date source code was saved (including century).
- HMC0066: "Y" if you want your last settings remembered, "N" if not.
HMC0067: <count> pages of Charts successfully generated in CMPRT02.
HMC0068: Index of Objects and/or Not Ref. List to be built from CMWKF04. Option: <option>
HMC0069: Table of Contents successfully generated in CMPRT01.
HMC0070: Index of Objects successfully generated in CMPRT03.
HMC0071: Not Referenced List successfully generated in CMPRT04.
HMC0072: Chart for seed <seed name> successfully generated in CMPRT02.
HMC0073: Chart Version <version> has completed successfully at <time> on <date>.
HMC0074: <PF key> is NOT a Valid PF Key. <PF key> ignored.
HMC0075: Answer "Y" for "Don't Explode" or "N" for "Explode".
HMC0081: No objects generated by wildcard types <types>. Processing terminated.
- HMC1001: "CHART for Natural" License will EXPIRE on <date>.
HMC1002: "CHART for Natural" License EXPIRED on <date>.
HMC1003: "CHART for Natural" License is INDEFINITE from <date>.
HMC1004: "CHART for Natural" License is >> INVALID <<.
- HMC2001: No Object Name supplied for Browse.
HMC2002: Use PF9 (Shape) to change window shape before using "Shift".
HMC2003: Line increment is not numeric.
HMC2004: Command <command> not Recognised.
HMC2005: Selected text value is blank. Choose another.
HMC2006: Only one previously scanned text value may be selected at a time.
HMC2007: Enter a value, choose a previous value, or PF3 to exit.

12 Installing CHART

CHART software is normally supplied on 3.5 inch diskette or on CD. The files required can be copied from the diskette or CD onto your machine using a standard **file copy** or **file transfer** routine. The software is held in two “binary” files on the diskette or CD, one file for Natural Object code and one file for Natural Source code.

12.1 Create target dataset

You must first create (allocate) the two target datasets on your mainframe. The target dataset must have a record format of FB (ie. Fixed length, Blocked records), one with a record length of 250 and one with a record length of 90.

If you use ISPF option 3.2 to create the datasets, select option ‘A’ (for Allocate new data set) or ‘M’ (for Enhanced data set allocation), provide the dataset name you require, and press enter. The following screen will allow you to enter space requirements, record format, and record length:

```

Data Set Name . . . . : <user>.chart.v31f.frompc.fb250 (or ..fb90)

Management class. . . . . (Blank for default management class)
Storage class . . . . . (Blank for default storage class)
  Volume serial . . . . . (Blank for authorized default volume)
Data class . . . . . (Blank for default data class)
  Space units . . . . . trks (BLKS, TRKS, CYLS, KB, MB or BYTES)
  Primary quantity. . . 15 (In above units)
  Secondary quantity. . 5 (In above units)
  Directory blocks. . . 0 (Zero for sequential data set) *
  Record format . . . . fb
  Record length . . . . 250 (or 90)
  Block size . . . . .
  Data Set Name type. . . (LIBRARY, HFS, PDS, or blank) *

```

12.2 Copy “binary” software file to target datasets

Copy each “binary” software file from the diskette or CD into the dataset you have just created. If you are using a standard “send” command, they should be:

1. **send x:\chart31f.bin ‘<user>.chart.v31f.frompc.fb250’**
2. **send x:\chart31f.src ‘<user>.chart.v31f.frompc.fb90’**

where “x” is the disk drive you are using.

If you are using a “windows” style file transfer package:

- choose the **send to host** option under the **transfer** menu;
- in the **from** area, select the file **x:\chart31f.bin** (where “x” is the disk drive you are using);
- in the **to** area, type the target dataset name **‘<user>.chart.v31f.frompc.fb250’**;
- for **transfer type**, choose the **binary** option (not the **text** option). If there is a ‘Convert from ASCII to EBCDIC’ option, ensure it is NOT selected. Ensure any ‘Add Carriage Return/Line Feed (CR/LF)’ option is also NOT selected;
- click on the **send** button;
- Repeat to send file **x:\chart31f.src** to dataset **‘<user>.chart.v31f.frompc.fb90’**.

12.3 Set up some "Natload" JCL

Where possible, "borrow" some JCL from an existing JCL deck that has been used previously to NATLOAD Natural objects (to the required database and FUSER file in your system). The format of these existing JOB, EXEC, STEPLIB, DDCARD, and CMPRINT cards should be copied to the new JCL deck for the "natload" step.

For IBM OS390 operating system, the required JCL should look something like this:

```

//*****
//* "NATLOAD" is a Natural Step to "load" the NATUNLD records from CMWKF01
//*****
//* PGM=... PARM=(... STEPLIB... and DDCARD... will be specific to your
//* environment. Copy them in from an existing "Batch Natural" JCL ...
//*****
//NATLOAD EXEC PGM=<mpm>,REGION=0K,PARM=('SYS=<batch>')
//*          ^^^^^^^^^^          ^^^^^^^^^^^^^^^^^^^^^^^^^^
//STEPLIB DD DSN=<mpm>.NATURAL.LOAD,DISP=SHR
//          DD DSN=SYS4.ADABAS.LOAD,DISP=SHR
//          DD DSN=SYS3.ADABAS.LOAD,DISP=SHR
//DDCARD DD DSN=<mpm>.ADABAS.DDCARD(???) ,DISP=SHR
//*****
//CMPRINT DD SYSOUT=*
//*****
//* CMWKF01 should have record format "FB" and record length 250 or 90.
//CMWKF01 DD DSN=<User>.CHART.V31F.FROMPC.FB250,DISP=OLD
//*****
//* CMSYNIN holds the NATLOAD command for "loading" from CMWKF01 ...
//CMSYNIN DD *
LOGON CHART
NATLOAD LOAD ALL * FM CHART REPLACE
/*
//*
//

```

For IBM VSE operating system, the required JCL should look something like this:

```
*****  
* Natural Step to "load" the NATUNLD records from CMWKF01  
*****  
* <load libraries>, <dev-xxx>, <dev-yyy>, <User>, <mpm>, <fnat>,  
* <fuser>, <fdic>, and <fsec> will be specific to your environment.  
* Copy them in from an existing "Batch Natural" JCL ...  
*****  
// LIBDEF *,SEARCH=(SAG,<load libraries>...),TEMP  
*  
// ASSGN SYSLST,<dev-xxx>  
// ASSGN SYSxxx,SYSLST  
*  
// ASSGN SYSRDR,<dev-yyy>  
// ASSGN SYSyyy,SYSRDR  
*  
* CMWKF01 should have record format "FB" and record length 250 or 90.  
// TLBL CMWKF01,'<User>.CHART.V31F.FROMPC.FB250'  
*  
// EXEC NATURAL,SIZE=(AUTO,64K),PARM='SYSRDR'  
OBJIN=R  
FNR=<fnat>,DBID=<mpm>  
FNAT=(<mpm>,<fnat>)  
FUSER=(<mpm>,<fuser>)  
FDIC=(<mpm>,<fdic>)  
FSEC=(<mpm>,<fsec>)  
IM=D  
MT=0  
MADIO=0  
MAXCL=0  
AUTO=OFF  
STACK=OFF  
/*  
ADARUN DBID=<mpm>,DEVICE=3390,MODE=MULTI  
/*  
LOGON CHART  
NATLOAD LOAD ALL * FM CHART REPLACE  
FIN  
/*  
/&  
* $$ EOJ  
/*
```

12.4 Run the “Natload” Job

Two JCLs have to be run: one for dataset <User>.CHART.V31F.FROMPC.FB250 and one for dataset <User>.CHART.V31F.FROMPC.FB90. Apart from the dataset name used for CMWKF01, the JCLs can be identical. The JCLs can be run as two steps in one job or as two separate jobs.

Submit the JCL(s) and check the messages when the resulting job(s) have completed. The NATURAL objects (held in the TSO dataset which you created earlier) will be “loaded” by this JCL job. 193 “cataloged” objects and 4 “source” objects should be loaded. Apart from the initial programs (called “CH” and “CHART”), all the other objects for the *CHART* software begin with “HMC”.

Below is an example summary from a successful NATLOAD of the ‘FB250’ dataset:

Statistical Report of Objects Processed		
	Saved	Cataloged
	-----	-----
Global Data Area	0	0
Local/Param Data Area ..	0	1
Maps	0	301
Help routines	0	9
Subroutines	0	0
Subprograms	0	34
Programs	0	27
Copycode	0	
Text	0	
Processor		0
Miscellaneous Objects ..	0	0
Total programming obj ..	0	372
Total Views Unloaded ...	0	
Total objects Read	0	372
Total Object Unloaded ..		372
*** NATUNLD has been terminated successfully ***		

Below is an example summary from a successful NATLOAD of the 'FB90' dataset:

Statistical Report of Objects Processed		
	Saved	Cataloged
	-----	-----
Global Data Area	0	0
Local/Param Data Area ..	0	0
Maps	0	0
Help routines	0	0
Subroutines	0	0
Subprograms	3	0
Programs	1	0
Copycode	0	
Text	0	
Processor		0
Miscellaneous Objects ..	0	0
Total programming obj ..	4	0
Total Views Unloaded ...	0	
Total objects Read	4	0
Total Object Unloaded ..		4
*** NATUNLD has been terminated successfully ***		

12.5 Copy Required SAG subprograms from SYSEXT

The *CHART* software uses some Software AG "user" subprograms. These subprograms need to be copied from library SYSEXT in the FNAT file to library CHART or library SYSTEM in the FUSER file.

Subprogram 'USR0010N' is always required by *CHART*, and one or more of 'USR1025N', 'NSCLI' and 'USR0050N' is required, depending which step library initialisation method is chosen.

The method requiring only subprogram 'USR1025N' is the default. This subprogram should therefore be copied from library SYSEXT on the FNAT file to library CHART or library SYSTEM on the FUSER file. (See section '13.7 Current Library and Step Libraries' on Page 90 for more details).

12.6 Make CHART a steplib for relevant users/applications

The *CHART* software is loaded into a library called CHART. In a NATURAL SECURITY environment, users who want to use *CHART* must have access to this library. Therefore, the library CHART must be defined as a steplib for all users and/or applications who will be using *CHART*.

In a non-NATURAL SECURITY environment, the 'NONATSEC' option for initialising CHART step libraries should be used. See '13.7.3 Initialising Step Libraries using option 'NONATSEC' on Page 91 for details.

12.7 Enter your License Information

One of the “source” objects supplied is the initial *CHART* program called “CHART”. To use the *CHART* software, the correct value for variable #LICENSE-CLIENT-ID must be set in this program and the program STOW-ed. The correct license setting will be supplied to your organisation along with the *CHART* software diskette or CD. If you need to extend a trial license or convert to a permanent license, please contact your local *CHART* software vendor and they will supply the required information.

12.8 Access to the default FUSER

It is no longer necessary to have a DDM available which points to your default FUSER. Previous versions of *CHART* needed some subprograms to be catalog-ed against such a DDM. *CHART Version 3.1F* finds the default FUSER in your environment and is able to read your NATURAL source code from that FUSER. If you have other FUSER files which *CHART* needs to access, please see section ‘13.2 Access to source code in other FUSER files’ on Page 89 for details.

13 Customising *CHART*

CHART software is supplied with some source code that may be customised to suit your site. The initial program for *CHART* is called 'CHART' and is provided as source code. It defines the default values for the various *CHART* parameters (eg. seed, exclusions, trace interval, don't explodes, step libraries) and also holds license information (eg. license duration, name and address of the licensed organisation, contact name and phone).

The default values for *CHART* parameters may be altered during installation (or at any later time) by amending and STOW-ing the program called 'CHART'. This program may also be copied and amended to provide differing default parameters for different users or sections in your organisation. The various parameters which can be amended are outlined later in this section.

13.1 Remembering previous *CHART* parameters

With *CHART Version 3.1F* each user's most recent *CHART* parameters can be 'remembered'. That is, whichever seed name, object exclusions, don't explodes, etc. were last used (on-line), the next time that user invokes *CHART* (on-line), those parameters will be re-displayed as the default settings.

The variable #SAVE-PARAMS in program 'CHART' controls how this feature works. By default, this variable is set to 'Y' which means each user's parameters will be saved each time they use *CHART*.

If a particular user does NOT want their parameters saved in this way, that user can change their own setting for #SAVE-PARAMS. To do this, the user should press PF12 (Parms) on the *CHART* menu and change 'Automatically Save Parameters?' to 'N', and then press PF6 (Updte). This process can be reversed at any time.

The #SAVE-PARAMS variable in program 'CHART' can also be set to 'N' which means that parameters will NOT be saved. While this is the default setting, any user can override the setting for themselves alone by following the process described in the previous paragraph.

Finally, the #SAVE-PARAMS variable in program 'CHART' can be set to 'D' which means that parameters will NOT be saved and users are NOT allowed to override this using PF12 on the *CHART* menu.

13.2 Access to source code in other FUSER files

The default FUSER file in any environment is automatically available to *CHART*. If your site has other FUSER files holding NATURAL source code, then *CHART* can be set up to access these files.

Three *CHART* subprograms are provided as source code and can be copied for each extra FUSER file that needs to be accessed. The subprograms are 'HMCINFOX', 'HMCLINEX', and 'HMCSUBRX'. A DDM for the required FUSER file needs to be available before these subprograms can be STOW-ed. The name of the relevant DDM must be entered into these subprograms:

```
1 FUSER VIEW OF <fuser-ddm-name>
2 SRCID
2 C*SRCTX
2 SRCTX(60)
```

The last character in the name of these subprograms (ie. 'X' in this case) should be added into the #ORDER-OF-SEARCHING variable in program 'CHART'. Alternately, the 'X' can be added to the 'Order of Searching' online by pressing PF5 on the *CHART* menu.

If more than one FUSER file needs to be accessed, copies of 'HMCINFOX', 'HMCLINEX', and 'HMCSUBRX' can be made for each extra FUSER file, altering the 'X' to any other character except 'F', 'G', 'U' or 'W'.

13.3 Order of searching FUSER files (and other repositories)

The order in which these FUSER files will be searched is defined in program 'CHART' by variable #ORDER-OF-SEARCHING. Amend the default value of this variable to say 'FHI' to search FUSER files accessed by first 'HMC...F', then 'HMC...H', and finally 'HMC...I'.

While using *CHART* on-line, PF5 on the *CHART* menu also allows this "order of searching" to be altered.

13.4 Default "Object Types" to be Excluded

The variable #EXCLUSIONS defines the 'object types' which will be excluded by default. Apart from the 'seed' name, this is the most commonly specified parameter when using *CHART* on-line or in batch. However, if it is advisable or preferable for certain object types to be almost always excluded at your site, then INIT-ialise this variable (in the program called 'CHART') to an appropriate value.

13.5 Default “Trace Interval”

The variable #TRACE-INTERVAL defines the default number of source code lines that will be searched on-line before a ‘status’ or ‘trace’ window is displayed. This variable is normally INIT-ialised to 5000 and can always be amended when drawing a *CHART*. However, you can set the ‘default’ to be any appropriate value (in the program called ‘CHART’).

13.6 Default “Don’t Explode” / “Don’t Show” / “Only Explode” Settings

Elements of the array #DONT-EXPLODE can be INIT-ialised (in the program called ‘CHART’) to values appropriate to your environment. It is normal for some almost universally referenced ‘utility’ objects to be defined as default values for #DONT-EXPLODE so you don’t have to enter them every time you draw a *CHART*.

13.7 Current Library and Step Libraries

The current library (defined in variable #LIBRARY) and the step libraries (defined in array #STEPLIB) are searched in order when a *CHART* is being built. Variable #LIBRARY will always default to the library from which you invoke *CHART* (ie. the same as system variable *LIBRARY).

You can choose how the step libraries are INIT-ialised (in the program called ‘CHART’) with the variable #INIT-OPTION. This variable can be set to ‘USR1025N’, ‘NSCLI’, ‘NONATSEC’, or ‘AUTOSTEP’, or left blank. These step libraries can always be overridden when drawing a *CHART* on-line or in batch.

13.7.1 Initialising Step Libraries using option ‘USR1025N’

If #INIT-OPTION is set to ‘USR1025N’, the default step libraries will be INIT-ialised using standard Software AG NATURAL subprogram ‘USR1025N’. Where NATURAL SECURITY is installed, this subprogram initialises the *CHART* step libraries to be the same as those defined (if any) for the user in NATURAL SECURITY. In a NATURAL SECURITY environment, ‘USR1025N’ can only read the steplibs. *CHART* cannot update the NATURAL steplibs, but only maintains its own copy. Only NATURAL SECURITY is allowed to set the steplibs. However, in a non-NATURAL SECURITY environment, ‘USR1025N’ is allowed to read and update the steplibs for the life of the session. If NATURAL SECURITY is not installed, use option ‘NONATSEC’. Subprogram ‘USR1025N’ will need to be copied from library SYSEXT on the FNAT file to library CHART or library SYSTEM on the FUSER file.

13.7.2 Initialising Step Libraries using option 'NSCLI'

If #INIT-OPTION is set to 'NSCLI' and if NATURAL SECURITY is installed, the default *CHART* step libraries will be INIT-ialised using standard NATURAL SECURITY subprogram 'NSCLI'. This subprogram initialises the *CHART* step libraries to be the same as those defined (if any) for the user in NATURAL SECURITY. Subprogram 'NSCLI' will need to be copied from library SYSEXT on the FSEC or FNAT file to library CHART or library SYSTEM on the FUSER file.

13.7.3 Initialising Step Libraries using option 'NONATSEC'

If #INIT-OPTION is set to 'NONATSEC' and if NATURAL SECURITY is NOT installed, the default *CHART* step libraries will be the step libraries (if any) defined when your environment was started/initialised.

Subprogram 'USR1025N' can be used in a non-NATURAL SECURITY environment to maintain step libraries. If the 'NONATSEC' option is used, *CHART* uses subprogram 'USR1025N' to read the current step library settings. If necessary, library CHART is added to these settings (so that *CHART* can be invoked from any library).

Subprograms 'USR0050N' and 'USR1025N' will need to be copied from library SYSEXT on the FNAT file to library SYSTEM on the FUSER file. Programs 'CH' and 'CHART' and subprograms 'HMC#INIT' and 'HMCFUSER' will also need to be copied from library CHART to library SYSTEM.

13.7.4 Initialising Step Libraries using option 'AUTOSTEP'

If #INIT-OPTION is set to 'AUTOSTEP', *CHART* will check whether NATURAL SECURITY is installed or not. If NATURAL SECURITY is installed, then *CHART* will initialise step libraries as if #INIT-OPTION was set to 'USR1025N' (as described in section **13.7.1 Initialising Step Libraries using option 'USR1025N'**). Otherwise, if NATURAL SECURITY is NOT installed, *CHART* will initialise step libraries as if #INIT-OPTION was set to 'NONATSEC' (as described in section **13.7.3 Initialising Step Libraries using option 'NONATSEC'**).

If NATURAL SECURITY is installed, then subprogram 'USR1025N' will need to be copied from library SYSEXT on the FNAT file to library CHART or library SYSTEM on the FUSER file.

If NATURAL SECURITY is NOT installed, then subprograms 'USR0050N' and 'USR1025N' will need to be copied from library SYSEXT on the FNAT file to library SYSTEM on the FUSER file, and programs 'CH' and 'CHART' and subprograms 'HMC#INIT' and 'HMCFUSER' will also need to be copied from library CHART to library SYSTEM.

13.7.5 Initialising Step Libraries directly in 'CHART'

If #INIT-OPTION is left blank, you can 'hard-code' the values of #STEPLIB (in the program called 'CHART') and they will be left untouched.

13.8 Default “Batch Invocation” Parameters

Batch NATURAL invocations using the standard Software AG program NATRJE are automatically detected and drawn by *CHART*. Some existing clients (who use “homegrown” pre-cursors to NATRJE) have had these “batch invokers” incorporated directly into the *CHART* software.

The variables #BATCH-INVOKER-n and #BATCH-PARAM-n define any site-specific “batch invocation” details to be used by *CHART*. These details allow “on-line => batch” links to be detected and drawn by *CHART*. When using *CHART* on-line, the current “batch invoker” settings can be viewed (and amended) by using PF7 on the initial *CHART* menu.

13.9 Default Settings for CHART characters and colours

On the *CHART* “seed” menu, PF12 allows the “down connector” and “cross connector” characters to be changed. Also, the colour settings for each object type and for all other information displayed in *CHART* can be altered. Any changes you make will only affect your current session. To permanently change any of these settings, you must EDIT and STOW the program called ‘CHART’.

Any of these characters and colours can be amended, but the variables which you are more likely to amend to suit your environment are:

- #DOWN-CONNECTOR (|),
- #CROSS-CONNECTOR (‘_’),
- #SEED-CV (‘Red’),
- #CONNECT-CV (‘Yellow’),
- #OUTLINE-CV (‘Neutral’),
- #SCREEN-SHAPE (‘Browse window starts as a Quarter size’),
- #INVERSE-VALUE (‘^’).

13.10 CHART and NATURAL Versions

The *CHART* software draws structure charts for applications written in any version of NATURAL (up to and including NATURAL Version 3.1.2). That is, *CHART* will draw structure charts for a NATURAL application consisting of any mix of NATURAL Version 1, 2 or 3 source code, whether in reporting mode or structured mode. If new calling constructions are introduced in future versions of NATURAL, *CHART* will be enhanced to detect and draw those object-to-object constructions.

The *CHART* software has been developed and compiled using NATURAL Version 2.2.8, but uses few special features of NATURAL Version 2.2. The software should execute successfully in most NATURAL Version 2.1 environments. However, for early NATURAL Version 2.1 environments, the variable #LICENSE-VERSION (in initial program ‘CHART’) may need to be amended from ‘3.1F’ to ‘2.1F’. This will force the *CHART* software to display on-line charts without ‘reverse video’ and without colour coding by object type (ie. using NATURAL Version 2.1 maps).

A

ASCII 75

B

batch .. 5, 17, 19, 28, 30, 38, 44, 46, 47, 48, 49,
50, 52, 53, 55, 56, 67, 69, 71, 76, 81, 82, 84
batch invoker..... 19, 48, 49, 84
binary 75
browse 5, 32, 39, 40

C

call..... 28, 29
calling method..... 28, 29, 36, 38, 52
CMPRT01 5, 50, 52, 53, 55, 68
CMPRT02 5, 50, 52, 53, 55, 68
CMPRT03 5, 50, 52, 53, 54, 55, 68
CMPRT04 5, 50, 52, 53, 54, 55, 68
CMWKF01 5, 17, 46, 47, 50, 52, 53, 54, 55,
59, 60, 61, 62, 63, 64, 76, 77
CMWKF02 .. 17, 46, 47, 50, 52, 53, 54, 55, 57,
59, 60
CMWKF03 50, 52, 54, 55, 67, 68
CMWKF04 .. 17, 46, 47, 50, 52, 54, 55, 59, 60,
64, 67, 68, 69, 70, 71
colour 24, 42, 84
connector..... 24, 45, 84
copycode 16
CSV 47, 54, 55, 60, 68, 69

D

date..... 5, 7, 24, 27, 37, 38, 52, 53, 54
DDM 79, 81
diskette 75, 79
dynamic..... 29

E

EBCDIC 75
exclude 4, 12, 13, 65
explode .. 16, 17, 29, 46, 47, 50, 59, 61, 62, 63,
64, 65
external..... 48

F

fetch..... 63
FUSER 18, 76, 79, 81, 82, 83

I

index of objects 5, 47, 50, 52, 53, 54, 56, 67,
68, 85
INPL..... 76
ISPF 75

J

JCL..... 50, 52, 55, 67, 68, 76, 77, 78

L

left..... 27, 31, 33, 36, 40, 42, 44, 54, 65, 82, 83
license 79, 80
lines..... 12, 15, 37, 40, 54, 58, 82
local 79

M

map 29

N

NAT1205 67
NATLOAD 76, 77
NATRJE 19, 48, 84
NATURAL 7, 25, 76
NATURAL 2.2 7
NATURAL 2.3 7
no source code 30
NONATSEC 79, 82, 83
not referenced 5, 52, 53, 54, 56, 67, 68
NSCLI..... 79, 82, 83

O

object type 4, 12, 13, 14, 15, 17, 24, 45, 46, 47,
50, 52, 65, 81, 84
object types 4, 12, 13, 14, 17, 46, 65, 81
on-line.. 4, 5, 26, 30, 39, 45, 48, 49, 80, 81, 82,
84
order of searching 18, 81

P

page limit 47, 50, 52, 54
parameters..... 6, 12, 25, 26, 48, 80
program 4, 5, 12, 16, 19, 28, 48, 49, 54, 66, 79,
80, 81, 82, 83, 84

R

right..... 27, 31, 33, 36, 39

S

Saving 26, 37
scan 41, 42
seed 5, 6, 12, 13, 14, 15, 17, 18, 19, 22, 24, 25,
26, 27, 29, 35, 45, 46, 47, 49, 50, 52, 53, 54,
55, 59, 60, 61, 63, 65, 66, 80, 81, 84
send 75
shape 12
sort..... 67
source code. 4, 5, 13, 15, 18, 30, 32, 37, 38, 39,
40, 58, 79, 80, 81, 82, 84
step library 79, 83
subprogram 16, 79, 82, 83
suppress..... 4
SYSEXT 7, 79, 82, 83
SYSTEM..... 79, 82, 83

T

table of contents... 5, 47, 52, 53, 54, 56, 67, 85
title 53, 55
top 27, 40, 41, 52

trace 6, 15, 35, 80, 82
transfer 75
TSO 78

U

unplaced objects 44, 45, 47, 54, 65
USR0010N..... 79
USR0050N..... 79, 83
USR1025N..... 79, 82, 83

W

work file..... 52, 53, 64

X

XREF 4, 5

Y

Year 2000 7

Z

zoom 33