SCRIPT USER GUIDE

Publication SCRIPT-1, July 15, 1981

PREFACE

SCRIPT makes the power of New England Digital's computer available to the creative musician, composer, or producer, as well as to the keyboard synthesist.

The convenience and versatility of SCRIPT is derived from its modular software architecture and highly sophisticated compiler design. Just pressing a key on the computer terminal rapidly converts the system from real-time Synclavier® II operation to the visual display and editing of the same musical information on the terminal.

SCRIPT MUSIC LANGUAGE

The powerful and easy to use SCRIPT musical notation system fills the needs of professional musicians. It is uncomplicated and similar in concept to standard musical notation. Yet it allows the musician precise control over many subtle aspects of music composition and performance.

You simply type in a composition's pitches, rhythm, and dynamics, and any other aspect of the piece. A motif may be repeated, played with different timbres, or subjected to the retrograde, transposition, or inversion operations of tonal music.

SYNCLAVIER® II COMPATIBILITY

The SCRIPT compiler quickly converts typed musical notation to real-time Synclavier® II performance pieces; the reverse compiler also quickly converts a live keyboard performance into a readable format for easy editing with the computer terminal. No loss of information or sound quality results from either conversion. Keyboard tracks can be combined with programmed tracks for complete flexibility.

EDITING

With SCRIPT, you can use the computer terminal to compose new pieces for performance on the Synclavier® II or to analyze, copy, edit or synchronize musical sequences originating from the Synclavier® II keyboard. In SCRIPT, tempo and timing may be expressed in seconds, frames, or musical beats.

POLYRHYTHMS

SCRIPT composition offers far more precise control of rhythm and dynamics than is possible in keyboard performance. You can easily develop an ostinato bass line or set up complex polyrhythms or straight rhythms which are extremely difficult, if not impossible, to perform accurately on a keyboard.

FILM AND VIDEO

The unique abilities of SCRIPT to synchronize musical beats to time or frame cues and to express tempo in frames offers new creative opportunities in film and video production. With SCRIPT, you can delay the start of a musical passage until a particular time or frame cue. Or, you can direct the system to change the tempo of the passage in order to synchronize it with both beginning and ending cues. Such tempo changes can be gradual or immediate, whichever you prefer.

COMPUTER COMPATIBILITY

SCRIPT provides for the easy transfer of Synclavier® II musical information to and from other computer systems.

USER GUIDE CONTENTS

The SCRIPT User Guide is directed particularly to the musician and contains background material, step-by-step instructions, and many musical examples. Once becoming familiar with the SCRIPT system through the User Guide, a musician can turn to the Reference Manual for quick review while entering compositions.

Using this Guide:

This Guide describes the SCRIPT system in its Level 1 release (July 15, 1981).

The first chapter presents the overall structure of the SCRIPT system. In the next chapter are step-by-step instructions on how to operate that system. The third chapter explains the SCRIPT compositional language. And, in the fourth chapter, more step-by-step instructions show you how to enter, edit and save SCRIPT compositions.

The two step-by-step chapters are designed to be read with the computer "on". You may, if you wish, read them together and then read the chapter on the language. Or, read the language chapter first.

The remaining chapters explain more advanced features.

The material in this manual is for informational purposes and is subject to change without notice.

New England Digital Corporation assumes no responsibility for any errors which may appear in this manual.

THE SCRIPT SYSTEM

You need the following hardware to operate the SCRIPT system:

1. Synclavier® II system with 40K, 48K, or 56K memory
2. Dual disk drives (5 1/4 inch "mini" or 8 inch "maxi")
3. Computer terminal

The following optional hardware will make your SCRIPT system even more versatile:

1. Winchester disk drive—offers faster compilation and greater storage
2. Remote disk drives ("maxi" systems only)—enable you to transfer compositions created in a "maxi" system onto a minidiskette for performance on Synclavier® II "mini" systems
3. Printer/Modem port—allows you to connect a hard-copy printer to the Synclavier® II computer or to transfer SCRIPT files to another computer
4. Hard-copy printer—allows you to print permanent copies of SCRIPT compositions if you are using a CRT terminal
5. Modem—allows you to connect the Synclavier® II computer to a remote computer, using the Printer/Modem port and telephone lines

You will receive the following diskettes with your SCRIPT system:

1. SCRIPT operating system diskette (In "maxi" systems, this diskette also contains the utility programs.)
2. SCRIPT user diskette
3. Utility Programs diskette ("mini" systems only)

And finally, you will receive the following documentation:

1. SCRIPT User Guide

   The SCRIPT User Guide is directed particularly to the musician and contains background material, step-by-step instructions, and many musical examples. Once having become familiar with the SCRIPT system through the User Guide, a musician can turn to the Reference Manual for quick review while entering compositions.

2. SCRIPT Reference Manual

   The SCRIPT Reference Manual is a compact, but complete description of the SCRIPT system. Experienced programmers may turn to this Manual immediately to learn the details of the system. All monitor commands are summarized as well as all SCRIPT statements, along with formats, value ranges, and default settings.

3. Options Setup Manual

   The Options Setup Manual includes information you may need when you first install the SCRIPT system, or if you add optional equipment in the future.

4. Utility Programs Manual

   The Utility Programs Manual includes information on formatting and duplicating diskettes, reorganizing the files stored on diskettes, making listings, and communicating with other computers.

5. Screen Editor Manual ("maxi" systems only)

   The Screen Editor Manual describes the use of the screen editor, a very fast and convenient way to enter and edit text on a CRT terminal.

   (The screen editor is included in all SCRIPT, MAX, and Scientific XPL/3 operating system maxidiskettes. For minidiskette systems, a special version of the screen editor which runs without the standard monitor may be available (at extra cost) for editing text files; it can play SCRIPT compositions or run XPL programs, but cannot handle compiled files in either system.)

The basic operation of the SCRIPT system is accomplished by four separate software modules: the *monitor*, the *SCRIPT compiler*, the *Synclavier® II real-time music performance system*, and the *reverse compiler*. Each module has a different function and is placed in computer memory at a different time. Yet they are all completely and automatically coordinated with one another. Several additional software modules, the *utility programs*, may be needed occasionally; you will use the *monitor* to place them in computer memory when they are needed. You control the entire operation by typing simple "commands" on the computer terminal.

The monitor

When you load the SCRIPT system diskette, the monitor is placed in computer memory. (NOTE: The word monitor has nothing to do with the high fidelity monitor used in audio systems.) At this time, the computer only responds to input typed on the computer terminal; it does not respond to input from the clavier unit. The monitor interprets the "commands" which you type on the terminal and coordinates the operation of the other programs. The monitor also enables you to enter SCRIPT compositions through the terminal, to edit compositions, and to store them on or recall them from the diskette.

The SCRIPT compiler

When you are ready to hear your composition, you type PLAY on the terminal. The SCRIPT compiler will then convert the composition into a Synclavier® II sequence, identical in form to sequences created through the Synclavier® II keyboard. The SCRIPT compiler also checks the composition for errors, and types messages on the terminal if any are found. During compilation, the computer will not respond to input from either the terminal or the clavier unit.

The Synclavier® II real-time performance system

After successful compilation of your composition, the Synclavier® II real-time performance system is placed in computer memory and the sequence is placed in the Synclavier® II memory recorder. At this time, the computer is controlled from the Synclavier® II buttons and keyboard and not from the terminal. You can modify the sequence by using the buttons and keyboard and store it permanently on diskette as a Synclavier® II sequence. You can return to the monitor program at any time by pressing the S key on the terminal.

The SCRIPT reverse compiler

You can also translate any Synclavier® II sequence (including a keyboard performance) into SCRIPT notation. When you press certain keys, the reverse compiler will translate the sequence in the memory recorder into a SCRIPT composition and return you to the monitor program. Moreover, there are several different formats for conversion, each designed for a specific use.

The utility programs

Several additional software modules are included in the SCRIPT system. These modules may be used to "configure" the system the first time it is set up or when optional equipment is added, to format new diskettes, to duplicate diskettes, to reorganize file storage on diskettes, and to establish communications with remote computers. Instructions for using them are in the Options Setup Manual, or in the Utility Programs Manual.

To get a feel for the operation of the SCRIPT system, run through the following step-by-step instructions. You will be learning "hands on" how to control the system through the computer terminal. You will also get a glimpse of the creative possibilities of SCRIPT musical notation.

If you have never used a computer terminal before, see the Options Setup Manual. It contains a chapter which explains the use of some of the special keys on the keyboard, such as CTRL (control) and ESC (escape).

NOTE: The SCRIPT system operates in a slightly different way when you are using a Winchester disk instead of dual floppy diskettes. Read "Using the Winchester" (Section 8 in this binder) before continuing with this manual.

LOADING

1. Turn on the terminal.
2. Turn on the computer.
3. Insert the SCRIPT *system* diskette into the *left-hand* disk drive. (NOTE: Do not remove the system diskette during operation of the SCRIPT system.)
4. Insert the SCRIPT *user* diskette into the *right-hand* disk drive.

   This diskette contains the sample compositions for you to work with and space for your own new compositions, as well as some Synclavier® II timbres.
5. Press the LOAD button on the computer.

   On the terminal will appear a release date and the word READY. At this point, the monitor is in computer memory. All the buttons and keys of the Synclavier® II are inactive, the lights will be off, and the synthesizers will be silent. If the system doesn't load, lights will remain lit and synthesizers will not be silent; in this case, see the Problem Section in the SCRIPT Reference Manual.

   The word READY appears whenever the monitor is ready for you to input commands.

THE MONITOR

You control the monitor by typing certain precise words, or *commands,* on the terminal. Each command must be completed by pressing the RETURN key before the monitor will initiate any action..The commands must be spelled correctly but they can be typed in either upper or lower case. If you type a word the monitor does not recognize, it will respond by writing "WHAT?" on your terminal. If that occurs, check your command for accuracy and retype it carefully. Should you discover an error before pressing RETURN, you can erase it one character at a time with the DELETE or RUB key or erase the entire line with the CTRL-X keys.

6. Now type the monitor command CATALOG and press RETURN.

   (Be sure to press RETURN after each and every command.) You will see listed the names of the following *user files*, or SCRIPT compositions, already stored on the diskette. (The files listed below contain examples used in explaining the initial release of SCRIPT. There may be additional files included in later releases to demonstrate new SCRIPT features.) Files may not necessarily be listed in this order.

TIMBRES	BACH13	BRAND3	POLLY
KEY	RHYTHM	PHRASE	ERRORS
DYNAMICS	TEMPO	SYNC1	SYNC2

   Each user file has a *file name*. File names can consist of up to eight letters or numbers. Remember, no distinction is made by the SCRIPT system between upper and lower case letters.

   (Some files on the user diskette are not listed by the CATALOG command. These are special files used to store and recall Synclavier® II timbres and sequences. The names of these "hidden files" begin with a period (".") and can be listed by the CATALOG ALL command.)

RECALLING A SCRIPT COMPOSITION FROM THE USER DISKETTE

7. Type OLD BRAND3 and press RETURN.

   OLD is a monitor command that creates a temporary copy of a particular user file stored on the diskette. This copy is called the *current file*. BRAND3 is the file name of a user file which contains a SCRIPT composition based on the first eight measures of Bach's Third Brandenburg Concerto.

   The OLD BRAND3 command makes the Brandenburg Concerto the current file. It can now be listed, played, and modified.

8. Type LIST

   The monitor command LIST causes the current file (in this case BRAND3) to be listed on the terminal. The list is divided into 19-line segments, the first of which will appear on the terminal. Press RETURN for successive 19-line segments. (NOTE: You can also press the LINEFEED key and one line at a time will be scrolled onto the screen. On the other hand, use the BREAK key to stop listing at any point.)

   If you are using a hard-copy terminal, such as a DECwriter, as your terminal, typing the monitor command NCRT once will cause the monitor to list the *entire* file every time you type LIST.

9. Type PLAY

   The monitor command PLAY causes the SCRIPT compiler to convert the SCRIPT composition in the current file into a Synclavier® II sequence. and places the Synclavier® II system in computer memory.

   After a brief delay during which compilation takes place, the piece will play.

SYNCLAVIER® II OPERATION

The Synclavier® II real-time performance system is controlled from the button panel and the keyboard. To experience SCRIPT's compatibility with the Synclavier® II, try the following:

a. Press STOP on the Synclavier® II RECORDER CONTROL panel.

b. Press START; the piece will play again.

c. Press TRACK 1 (it will flash), and the first violin part will play individually.

d. Press TRACK 1, then TRACK 2, and the second violin part will play individually.

e. Press STOP.

f. Try replacing one of the timbres: Press TIMBRE BANK 8, then the SMT key, then TRACK 1, then TIMBRE ENTRY 4, and finally START.

As you can see, you can do the same musical manipulations with the sequences created by the SCRIPT musical notation system as with those generated on the Synclavier® II keyboard. (NOTE: There is one difference from normal Synclavier® II operation. Timbres and sequences must be stored on and recalled from the *right-hand* disk drive, not the left. The DRIVE SELECT button is automatically "on". All SCRIPT user diskettes contain Synclavier® II timbre banks.

Only a few terminal keys are active during Synclavier® II operation. Each terminates real-time performance and returns the monitor program.

10. Type S on the terminal when finished experimenting with BRAND3 in real time.

    This stops Synclavier® II operation and restores the monitor. BRAND3 is still your current file.

CONVERTING A SEQUENCE INTO SCRIPT

Both the timbres and the notes of a sequence in the memory recorder may be translated into SCRIPT notation. Instead of typing S, you could have made use of the reverse compiler.

1. Type PLAY again.
2. After the composition starts to play, press the PF1 key on the right-hand side of your terminal. Or, if your terminal does not have these special keys, press the ESC key and then the 1 key.

   You will see the following words on the terminal screen while the sequence is being translated:

```
Conversion Program now running
```

Once the translation is completed, you will see this message:

```
Conversion successful—Current file now called "BRAND3*".
READY
```

The monitor is restored and your current file now contains the converted sequence. Notice that the name of your current file has been automatically changed by adding an asterisk. This is to prevent confusion since the converted sequence has *replaced* your original current file. (The copy of BRAND3 which is *saved* on the diskette has not been changed.)

3. Type LIST

   On your terminal will be listed SCRIPT timbre definitions, followed by notelists. The format of the timbre definitions is described in detail in the chapter "Timbre Definition". The notelists each consist of a list of notes. Each note appears on a separate line in a format which includes its starting time, pitch, duration, and optional volume and timbre modification percentages. The format of the notelists is described in detail in "SCRIPT Computer Music Format".

   You can play this composition again by typing PLAY. You can also alter it on your terminal and store it on diskette in this form. This will be covered later.

   There are several other formats into which you can translate a sequence. Some are designed for easier editing and others for expressing starting times and durations in frame numbers. Details will be covered in later chapters.

MORE EXPERIMENTATION

To show you more of SCRIPT's musical capabilities, we have prepared another composition. This piece, named POLLY, is not a finished work. Rather, it is an absolutely accurate polyrhythmic tool. Eight separate rhythmic lines loop independently.

1. Type OLD POLLY

   The user file POLLY becomes the current file. You can LIST it if you want.

2. Type PLAY

   When POLLY starts playing, eight tracks with different rhythms will play at once. Experiment in real time by playing together various combinations of the tracks. Use two or three tracks as a polyrhythmic bass line while you play on the keyboard. Or transpose single tracks.

In this chapter, you have encountered the monitor commands CATALOG, OLD, PLAY, and LIST. Other monitor commands having to do with editing and saving compositions will be explained in the chapter "More About the Monitor".

To turn off your system, first remove your SCRIPT system and user diskettes from the disk drives and replace them in their jackets; then, turn off the computer and the terminal. For the next chapter, which explains the SCRIPT music language, you will not need the computer.

This chapter explains the close relationship between SCRIPT musical notation and standard notation. It describes the basic format, the various ways of specifying and modifying notes, and the means of accessing the Synclavier® II performance controls.

More advanced concepts, such as musical motifs with tonal transformations and complex overall tempos with accelerandos and ritardandos, are described in later chapters. Also reserved for later discussion are the features of SCRIPT particularly relevant to film and video work, such as timing in terms of frames and the synchronization of sequences to precise timing cues.

SCRIPT MUSICAL NOTATION

SCRIPT musical notation is a fast, convenient way of composing, editing, and displaying musical scores for the Synclavier® II on a computer terminal.

In standard musical notation, scores are written graphically for quick visual comprehension and interpretation by the performing musician. In a computer music language, however, every nuance of musical performance must be spelled out explicitly. A graphic presentation that resembles the notes on a staff of standard music notation would become cluttered with other information. For professional use, such a system would be clumsy and inflexible.

The SCRIPT notation system retains the basic note description and linear organization of standard notation. But, by using an uncomplicated, alphanumeric format for computer input, SCRIPT takes full advantage of the text manipulation capability of the computer terminal.

BASIC FORMAT

Each line of a SCRIPT composition consists of a line number, a SCRIPT word identifying a particular aspect of the composition, such as pitch, tempo, key signature, or articulation, and one or more numeric values or expressions separated by spaces. The formats of the various SCRIPT lines will be explained in the following sections.

Notelists

A SCRIPT *notelist* consists of all the notes to be played on one track in the memory recorder. The first line of a notelist begins with the word NOTELIST; the last line of the notelist consists of the word END. In between are the actual notes. Since there are 16 tracks in the Synclavier® II memory recorder, there can be 16 notelists in a SCRIPT composition.

Following the word NOTELIST is an expression identifying the timbre to be used on that recorder track. The expression takes the form

```
    USING Bank - Entry
```

and indicates a timbre stored in a timbre bank on the user diskette.

For example,

```
100 NOTELIST USING 3-2
```

begins a notelist to be played with the timbre stored in Bank 3, Entry 2.

Pitches

As in standard musical notation, the notes in a melody are listed one after the other from left to right. Instead of a single graphic symbol on a staff, the pitch and rhythmic values of a note are listed on separate lines of text. The *pitch line,* which must come first, begins with the letter P followed by the pitches. Pitches are indicated by the conventional letter names A through G. You separate one pitch from the next by typing one or more spaces.

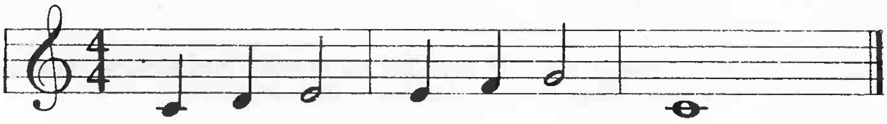
Rhythms

The *rhythm line,* which must come immediately after the pitch line, begins with the letter R, followed by the rhythmic values associated with the pitches in the pitch line. Rhythmic values can be indicated in a variety of ways. In simplest form, they are listed as single numbers, each of which represents a fraction of a whole note. For example, 1 represents a whole note; 2, a half note; and 4, a quarter note. You separate one rhythmic value from the next by typing one or more spaces.

Examples

In the following example, you will see a simple musical passage written in standard notation and in SCRIPT notation.

Example 1



```
100 NOTELIST USING 8-1
110  P    C  D  E    E  F  G    C
120  R    4  4  2    4  4  2    1
130 END
```

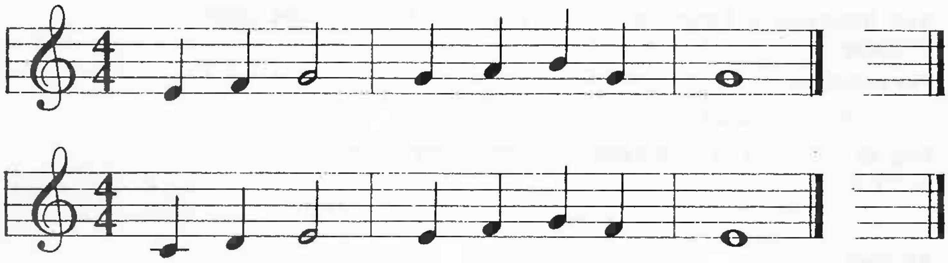
When this sequence is played, the notes will be played with the timbre in TIMBRE BANK 8, ENTRY 1.

Note the vertical arrangement of each note. There must be the same number of rhythmic values as there are pitches in the line above. Although you don't have to align the rhythmic value perfectly with the pitch of the note, it is easier to read through your composition if you do.

There are no measures in SCRIPT. You can list any number of notes on a set of pitch and rhythm lines. If you run out of room on one set of lines, simply continue with another set. Unlike the measures of standard notation, division of a notelist into many lines does not imply that any stress is to be applied to particular notes.

The following composition in two parts further illustrates the basic notelist format.

Example 2



```
100 /* An example of two notelists,
110    each "measure" is written on separate pitch and rhythm lines */
120
130 NOTELIST USING 8-1
140
150   P      E  F  G
160   R      4  4  2
170
180   P      G  A  B  G
190   R      4  4  4  4
200
210   P      G
220   R      1
230
240 END
250 NOTELIST USING 8-1
260
270   P      C  D  E
280   R      4  4  2
290
300   P      E  F  G  F
310   R      4  4  4  4
320
330   P      E
340   R      1
350
360 END
370
```

Note how a musical line can be divided into several lines of SCRIPT. Observe also the use of the symbols

/* . . . . */

Anything enclosed within these symbols is called a *comment* and is ignored by the SCRIPT compiler. Note too the use of *blank lines;* these are also ignored by the compiler and are used to make a composition more readable and less cramped in appearance.

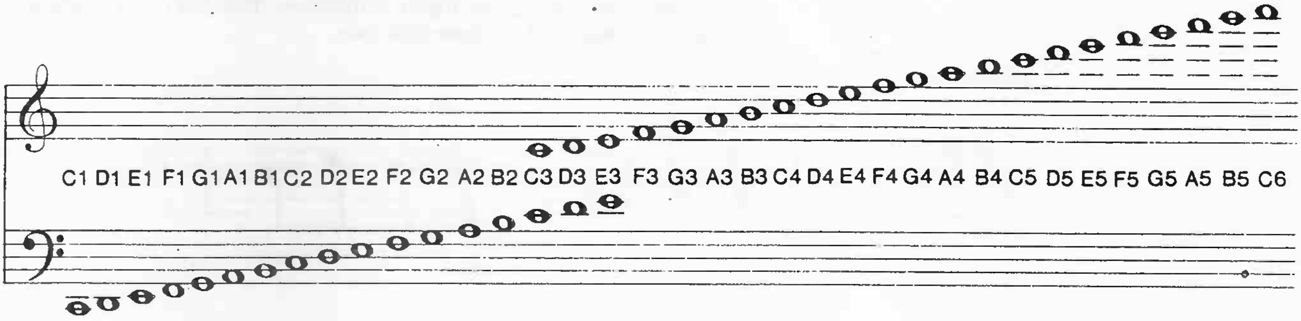
The two notelists will play simultaneously on TRACK 1 and 2.

MORE ABOUT PITCH

As you have seen, you indicate natural pitches by the conventional letter names A through G. To indicate a sharp, type the symbol "#" immediately after the pitch (with no space between); to indicate a flat, use the letter F in the same way. Unlike standard notation, accidentals do not affect subsequent notes.

To indicate octave, type a number immediately after the pitch. Octave numbers do affect subsequent notes; you don't have to specify octave for every single note. If you do not follow a pitch with an octave number, the note will be played in the same octave as the preceding pitch. If you omit octave number from the composition entirely, all notes will automatically be played in the octave above middle C (as in the earlier examples).

Below are the SCRIPT pitch values which correspond to the notes of the staff.


C1 D1 E1 F1 G1 A1 B1 C2 D2 E2 F2 G2 A2 B2 C3 D3 E3 F3 G3 A3 B3 C4 D4 E4 F4 G4 A4 B4 C5 D5 E5 F5 G5 A5 B5 C6

The KEYSIG Statement

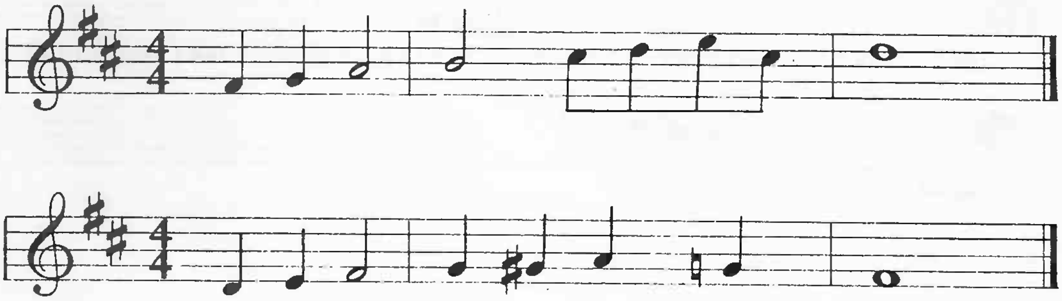
To simplify the entering of a composition with many sharps or flats, SCRIPT provides the KEYSIG statement. You can use this statement to establish an overall key signature for all succeeding notes. If you want the composition to be played in one key all the way through, a single KEYSIG statement inserted before the first notelist will establish all sharps or flats for all the notes. (NOTE: Accidental naturals can be indicated by the letter N immediately following the pitch.)

Follow the word KEYSIG by a legitimate Western scale major key, such as F#, C, or BF (B flat). A minor key signature is established by entering its relative major key. For example, for F# minor, use the key of A. (A list of keys may be found in the SCRIPT Reference Manual.)

You can change key signatures by inserting another KEYSIG statement. Remember, however, KEYSIG statements establish the key signature for all following notes, including notes in subsequent notelists.

The following musical example (on your SCRIPT user diskette under the file name KEY) illustrates the KEYSIG statement, accidentals, and octave changes.

Example 3



```
100 KEYSIG D
110 NOTELIST USING 8-4
120
130   P    F  G  A    B  C4  D  E  C      D
140   R    4  4  2    2  8   8  8  8      1
150
160 END
170
180 NOTELIST USING 8-4
190
200   P    D3  E  F    G  G#  A  G       F
210   R    4   4  2    4  4   4  4       1
220
230 END
```

Note the octave change at C4 on line 130; all succeeding notes are played in octave 4. For that reason, the first pitch in the second notelist, the D on line 200, must be followed by a 3.

MORE ABOUT RHYTHM

In SCRIPT all rhythmic values are based on beats. When you type PLAY, the SCRIPT compiler translates the beats of the composition into an actual time sequence according to a tempo indication. In our first examples, we did not specifically indicate tempo. We allowed the SCRIPT compiler to use a *default* tempo of 120 quarter notes per minute.

The TEMPO Statement

There are two ways to specify tempo in SCRIPT. The simplest way is with the TEMPO statement, inserted at the beginning of the composition and at any later point, whenever you want the tempo to change. Each TEMPO statement affects the duration of every note that follows until another such statement is entered. TEMPO statements can be used in simple, single notelist compositions and when you want to set up different tempos in different notelists.

The second way, described in the late chapter "Tempo Control", is with a group of tempo control statements entered at the beginning of the composition. Complex tempos, with accelerandos and ritardandos, can be set up which will affect corresponding beats in all notelists.

A TEMPO statement can take one of three forms. All three begin with the word TEMPO. In the first form, the word TEMPO is followed by an expression in which you specify beats per minute and the rhythmic value for the beat. The expression takes the form:

```
    rhythmic value = beats per minute.
```

The default TEMPO, for example, is equivalent to

```
    TEMPO 4 = 120
```

You can base the beat of your composition on any rhythmic value.

In the second form, you express tempo in terms of period, or time, instead of beats. After the equal sign, enter a time in seconds followed by the word SECOND.

In a composition with this TEMPO statement

```
100 TEMPO 4 = .35 SECOND
```

every quarter note would last exactly .35 of a second. (NOTE: You must express tempo in seconds if you plan to specify independent loops in your composition.)

In the third form, you express tempo in terms of "frames". After the equal sign, enter a number and the word FRAMES. For example, you might enter this TEMPO statement:

```
100 TEMPO 4 = 12 FRAMES
```

Each quarter note would then be computed to last as long as 12 frames. The actual duration of the quarter note is determined by the frame rate. (In SCRIPT there is a default frame rate of 24 frames per second. Thus, the above TEMPO statement would result in quarter notes that lasted half a second.)

The FPS Statement

You can, however, specify any frame rate by using the FPS (frames per second) statement. Use the word FPS, followed by any rate from 1 to 299.9. The following statement

```
100 FPS 30
```

would establish the frame rate of 30 frames per second used in American video. The FPS statement, if used, must be entered before the TEMPO statement.

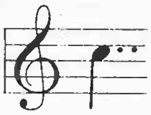
Rhythmic Values

When a rhythmic value is given as a single number, this number represents a fractional division of a whole note. Thus, 8 yields an eighth note, 16 a sixteenth note and 5, a quintuplet quarter note. Any number up to 64 can be used.

Rhythmic Values As Fractions

You can express any fraction of a whole note. A note with a rhythmic value of five eighths of a whole note can be written as 5/8. Even fractions representing complex rhythms such as 11/17 may be used. (Note that the rhythmic value 4 is equivalent to, and may be written as, 1/4; the same is true for all rhythmic values represented by a single number.)

Example 4



```
P     C4
R  7/16
```

You can also enter notes of longer duration than a whole note. Thus, you can represent a note with a rhythmic value of four tied whole notes by the expression 4/1.

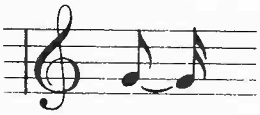
The standard rhythmic values can also be represented alphabetically as follows: W for whole note, H for half note, Q for quarter note, E for eigth note, S for sixteenth note, T for thirty-second note, and X for sixty-fourth.

Also available are simple symbols for the tied notes, triplet notes and dotted notes of standard music notation. With these conveniences, the same rhythmic values may be written in several different ways. We will explain all these conventions and then present an example which incorporates most of them.

Tied Notes

Rhythmic values can be added. If you insert a plus sign between rhythmic values, the SCRIPT compiler will add them together into *one* note of combined duration. Remember, you are creating one note. Be sure to include only one pitch on the line above for a tied rhythmic value.

Example 5

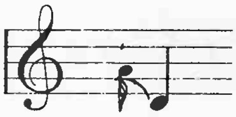


```
P  G
R  8+16
```

Grace Notes

Rhythmic values can also be subtracted. This feature provides a convenient method of implementing grace notes. First you list the rhythmic value of the grace note. Then you subtract that value from the main note.

Example 6

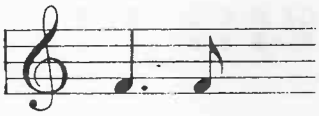


```
P     A   D
R     32  4-32
```

Dotted Notes

The SCRIPT *dotted note* corresponds to the dotted note in standard music notation. It multiplies the rhythmic value by one and one half. You can dot any rhythmic value. You can also double and triple dot.

Example 7

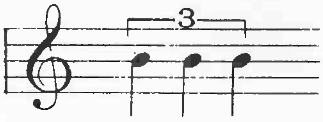


```
P     F   F
R     4.  8
```

Triplet Notes

Typing T after a rhythmic value creates a *triplet note.* The T multiplies the rhythmic value by 2/3.

Example 8

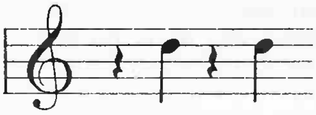


```
P     B   B   B
R     4T  4T  4T
```

The Rest

You can easily enter a *rest* in SCRIPT. Use the letter R or a period (".") in the appropriate position on the pitch line with the rhythmic value of the rest on the line below.

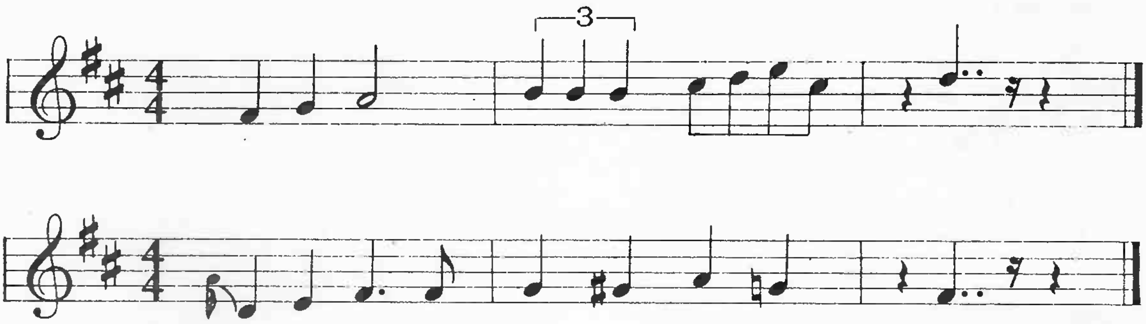
Example 9



```
P     R D4. D
R     4 4 4 4
```

The following example (in user file RHYTHM on your user diskette) adds a TEMPO statement, a rest, and grace notes, dotted notes, tied notes, triplet notes and fractional notes to the demonstration composition.

Example 10



```
100 KEYSIG D
120 TEMPO 4=126
130
140 NOTELIST USING 8-5
150
160 P  F G A    B  B  B  C4 D E C   R  D
170 R  4 4 4+4  4T 4T 4T 8  8 8 8   4  4..
180
190 END
200
210 NOTELIST USING 8-5
220
230 P  A3  D    E F  F   G G# A  G    R  F
240 R  32  4-32 4 4. 8   4 4  4  4    4  7/16
250
260 END
```

Note that the double-dotted quarter note can be entered in two ways: as 4.. and as 7/16. The rhythmic values are the same. Note also that the grace note is "on the beat."

ADDITIONAL FEATURES

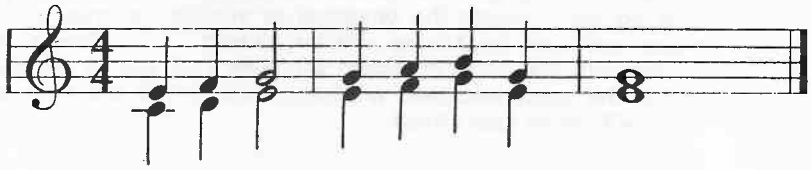
There are a few special symbols which can be used to enter chords, slurs, and repeating notes.

Simultaneous Notes

In SCRIPT you can use *brackets* to enter simultaneous notes or chords. Pitches enclosed within the same set of brackets will start together rather than sequentially. By using brackets, the composition in Example 2 could be written in the following single notelist. If so, all notes would be played on one memory recorder track (in Example 2, they are played on two tracks).

Your timbre must be polyphonic for multiple notes on a single memory recorder track to sound simultaneously (see the section on Keyboard Polyphony Control in the *Synclavier® II Instruction Manual, Revision D,* pages 2 and 3).

Example 11



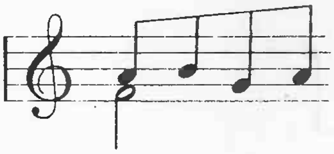
```
100 NOTELIST USING 8-1
120
130    P    [C  E]  [D  F]  [E  G]
140    R     4  4    4  4    2  2
150
160    P    [E  G]  [F  A]  [G  B]  [F  G]
170    R     4  4    4  4    4  4    4  4
180
190    P    [E  G]
200    R     1  1
```

Note that each pitch within the brackets must have its own rhythmic value in the rhythm line. Because the rhythmic values for the notes are the same, they will play for the same length of time.

Using separate notelists allows you to specify different timbres in the two parts. Brackets are especially useful for entering the occasional chord in a generally monophonic melody line.

It is also possible for the notes within brackets to have different rhythmic values:

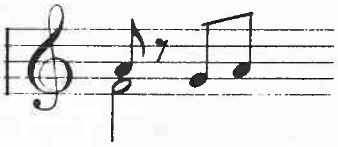
Example 12



```
P  [F  A]  B  G  A
R   2  8   8  8  8
```

The F and A start together. The F is given a half-note rhythmic value and the A is given an eighth-note rhythmic value. SCRIPT will cause the note following the brackets to be played immediately after the last note within the bracket. In this case, eighth-note B will begin immediately after the A, the last note within the bracket. Thus, the next three notes, B, G, and A, will sound on top of the F.

Example 13



```
P  [F  A]  R  G  A
R   2  8   8  8  8
```

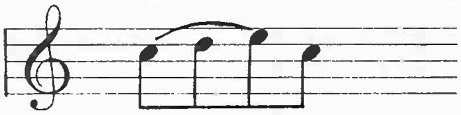
In this example, a rest has been entered after the bracket. This rest prevents the G from being played immediately after the A; the G will begin in the middle of the F.

Note that brackets are only used on the pitch line and are not used on the rhythm line.

Slurs

In SCRIPT, you can use *braces* to implement pitch changes that do not retrigger the envelope or vibrato generators. The notes enclosed by braces will be slurred. If the timbre has portamento, the notes enclosed will glide from one to the next. If not, the notes enclosed will jump from one to the next, but there will be no new attack.

Example 14



```
P    {C4D  E}  C
R      8  8  8   8
```

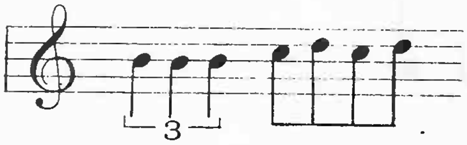
(NOTE: You cannot use both brackets and braces around the same notes. If you want to slur chords, you will have to use more than one notelist.)

Repeating Values

You will find the entering of certain kinds of musical passages less time-consuming if you make use of *parentheses* around pitches, rhythmic values, and sequences.

In this way you can efficiently type in any repeated pitches or rhythmic values (or repeated percentages on the articulation, volume or timbre lines to be described shortly), and any repeated sequences of such values as well.

Example 15

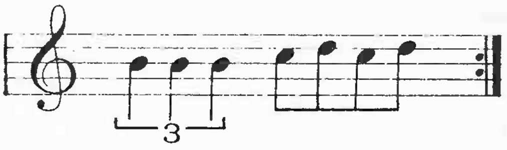


```
P  (B)3    (C4  D)2
R  (4T)3   (8)  4
```

(B)3 is the same as B B B; (C D)2 is the same as C D C D; (4T)3 is the same as 4T 4T 4T; and (8)4 is the same as 8 8 8 8.

You can also use multiple levels of parentheses.

Example 16

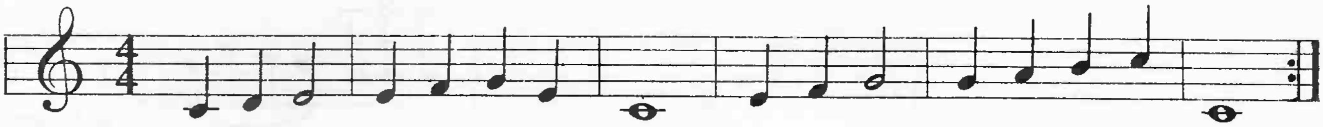


```
P  ((B)3   (C4  D)2)2
R  ((4T)3  (8)  4   )2
```

You may want to repeat a sequence longer than a line (the maximum number of characters allowed on a line is 128). To repeat such a sequence, you may break a pitch or rhythm line into multiple lines. You can then place the opening parentheses on one line and the closing parentheses on another.

Multiple pitch lines in row are equivalent to one long pitch line. The same is true of multiple rhythm lines. When using multiple lines, make sure that the total number of pitches matches the total number of rhythmic values in the set of lines.

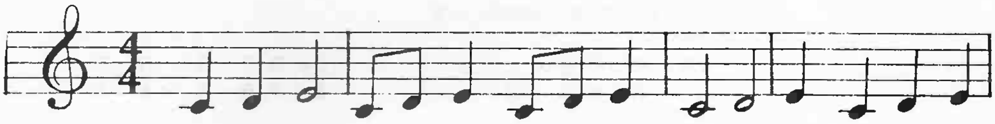
Example 17



```
P  (C  D  E  E  F  G  E  C
P    E  F  G  G  A  B  C4  C3) 2
R  (4  4  2  4  4  4  4  1) 4
```

This feature can be used to repeat a sequence of pitches, each time with different rhythmic values.

Example 18

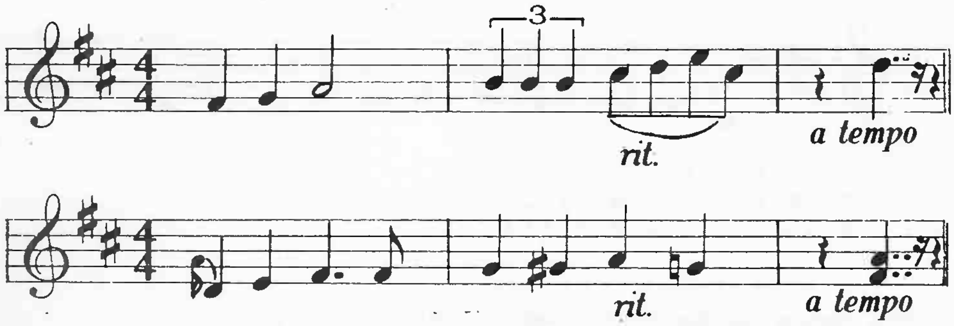


```
P  (C  D  E)  5
R    4  4  2
R  (8  8  4)  2
R    2  2  4
R    4  4  4
```

In the same way, you can extend braced and bracketed pitches over multiple lines.

The following composition, stored in user file PHRASE on your user diskette, adds simultaneous notes, slurs, repeating pitches and rhythms, and a *ritardando* to our demonstration composition.

Example 19



```
100  KEYSIG D
120  TEMPO 4=126
130
140  NOTELIST USING 8-5
150
160   P    F G   A      B    B   B    {C4   D  E}   C    R    D
170   R    4 4   4+4      (4T)3        8    7  6   5     4    7/16
180
190  END
200
210  NOTELIST USING 8-5
220
230   P    A3  D       E  F   F     G  G#   A     G      R    [F     A]
240   R    32  4-32    4  4.  8     4   4   8+7   6+5    4    7/16  7/16
250
260  END
```

The rather primitive *ritardando* has been entered in this way to show the use of nonstandard rhythmic values. There are more elegant methods of creating *ritardandos* presented in the later chapter on tempo control.

ARTICULATION, TIMBRE MODIFICATION AND VOLUME

There are three more ways to control notes in SCRIPT musical notation: articulation, volume, and timbre modification. Unless you specify otherwise in the notelist, notes will be played with 100 percent "articulation" (that is, the full length of their rhythm value); they will be played at 100 percent volume; and they will be interpreted with 100 percent of any "real-time effects" which are in effect on the timbre.

You can insert articulation, volume and timbre control lines under the pitch and rhythm lines to change these percentages on a precise, note by note basis.

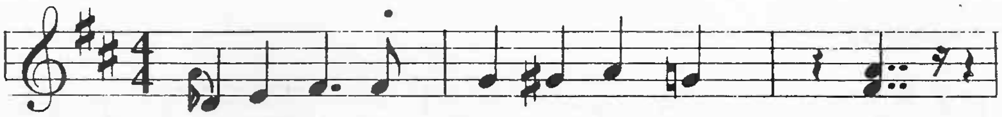
Each line starts with a letter (A, T, or V), followed by a list of percentages (from 0 to 100.0), one for each note in the set of pitch and rhythm lines above.

Articulation Line

In SCRIPT, articulation is defined as the percent fraction of the full rhythmic value of the note, with a full duration note being 100 percent or "legato".

You have complete control over articulation.

Example 20



```
P   A3  D      E   F   F   G   G#  A    G    R   [F    A]
R   32  4–32   4   4 . 8   4   4   8+7  6+5  4   7/16  7/16
A   100 50     50  50  30  60  60  60   60   0   100   100
```

The articulation line starts with the letter A and contains one percentage indication for each note in the pitch and rhythm lines above. A 100 causes the note above to be played full rhythmic value; 30 creates a staccato.

Remember, you need not include an articulation line for every set of pitch and rhythm lines in a composition. Or for any. But if you do include an articulation line, you must specify a percentage for every note, and rest, on the lines above. (Of course, for the articulation percentage under a rest you can choose any number. Also, for slurred notes, all articulation percentages except that for the last one within the braces will be ignored.)

Changing the Articulation Default

As we stated above, the articulation for notes wihout an articulation line is 100.0. You can change this default by inserting a statement starting with the word ARTICULATION, followed by a percentage. For example, the statement

```
100 ARTICULATION 75.0
```

causes every subsequent rhythmic value to be multiplied by 75 percent, *unless* a different percentage is specified for the rhythmic value on an articulation line. An articulation specified for a note will override a default.

Scaling All Articulations

You can quickly scale all articulations, both default and those specified note by note, by inserting a statement starting with the words ARTICULATION MULTIPLIER, followed by a percentage number (up to 1000.00 percent). For example, the statement

```
110 ARTICULATION MULTIPLIER 150
```

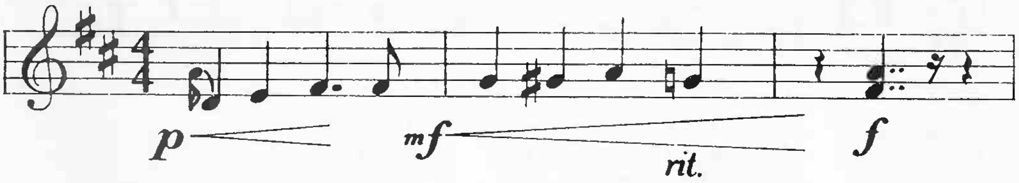
would cause all articulation percentages to be multiplied by 150 percent (that is, a factor of 1.5).

If the ARTICULATION MULTIPLIER causes a duration to exceed the rhythmic value of the note, the duration will be *truncated* to equal the full rhythmic value of the note.

The Volume Line

On the Synclavier® II, overall volume may be controlled by the pedal. In SCRIPT, you can specify note by note volume changes in a volume line. You can easily enter crescendos or decrescendos. Or accent certain notes to establish rhythmic patterns. (Of course, you can never create a volume louder than 100 percent.) Used in combination with the special slurring feature, described above, the volume line can be used to make a single note swell and fade away.

Example 21



```
P  A3   D     E   F   F    G   G#  A    G     R   [F    A]
R  32   4–32  4   4.  8    4   4   8+7  6+5   4   7/16  7/16
A  100  50    50  50  30   60  60  60   60    0   100   100
V  20   20    25  25  30   30  40  50   50    0   60    60
```

Observe how the percentages increase in the volume line. Observe also that values must be entered under the rest; these values are arbitrary.

Changing the Volume Default

In order to change the default percentage for volume, use the word VOLUME followed by a percentage. All notes following such a statement will be played at that percentage volume, unless you specify otherwise on a volume line.

Scaling All Volumes

As with articulation and timbre modification, you can quickly scale all volume percentages, both default and those specified for notes. Use the words VOLUME MULTIPLIER, followed by a percentage, which will multiply all volume percentages that follow.

If the VOLUME MULTIPLIER causes a volume percentage to exceed 100.0 percent, that percentage will be truncated to the maximum volume, 100.0.

Timbre Line

In normal Synclavier® II operation, the REAL-TIME EFFECTS panel and pedal allow you to change certain timbre qualities during performance. You "patch" the desired qualities into the timbre and operate a pedal to change the sound while you are playing.

In SCRIPT, you can use the REAL-TIME EFFECTS in a different and more precise way with a timbre line.

The timbre line starts with a T, followed by a list of percentages which will override the default value of 100.0 percent set for all real-time effects. All notes will be played as if you had the pedal all the way down unless you specify otherwise in a timbre line.

As with the articulation line, the use of a timbre line is entirely optional. But if you do insert a timbre line, you must specify a percentage for every note, and rest, on the pitch and rhythm lines above.

Changing the Timbre Default

In order to change the default for timbre modification from the preset setting of 100.0 percent, use the word TIMBRE followed by a percentage. All notes will be given this percentage of real-time effects, unless otherwise specified on a timbre line.

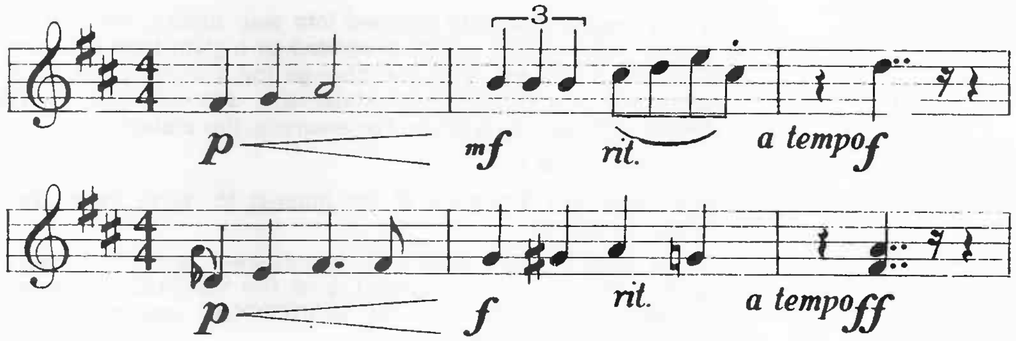
Scaling All Timbre Percentages

As with articulations, you can quickly scale all timbre modification percentages, both default and those specified for a note. Use the word TIMBRE MULTIPLIER followed by a percentage which will multiply all timbre percentages that follow.

If the TIMBRE MULTIPLIER causes a timbre modification percentage to exceed 100.0 percent, the percentage will be automatically truncated to the maximum timbre modification percentage, 100.0.

The following example, stored in user file DYNAMICS on your diskette, adds articulation, volume and timbre lines and a timbre default statement to our demonstration composition.

Example 22



```
100 KEYSIG D
110 TEMPO 4=126
120 TIMBRE 75
130
140 NOTELIST USING 8-5
150
160 P   F   G   A   B   B   B   {C4  D    E}   C   R  D
170 R   4   4   2   (4T)3       8    7    6    5   4  7/16
180 V   20  25  30  40  45  50  70   75   80   60  0  80
190 A   60  60  60  30  30  30  100  100  100  25  0  100
200
210 END
220
230 NOTELIST USING 8-5
240
250 P   A3   D      E   F   F   G   G#  A    G    R  [F     A]
260 R   32   4–32   4   4.  8   4   4   8+7  6+5  4  7/16   7/16
270 V   20   20     25  25  30  30  40  50   50   0  60     60
280 A   100  50  50 50  30  60  60  60  60   60   0  100    100
290 T   50   75     75  75  50  50  50  75   50   0  100    100
300
310 END
```

Line 120 establishes a timbre default of 75 percent. The timbre percentages specified in line 290 will override this default. Since there is no such timbre line in the first notelist, all the notes in it will receive 75 percent real-time effects.

USING SCRIPT TO SET SYNCLAVIER® II PERFORMANCE CONTROLS

There are several SCRIPT statements which give you direct access to Synclavier® II performance controls, such as the CLICK RATE or the LOOP function. You can use any or all of these in your SCRIPT composition. The TRANSPOSE, GLIDEFROM and independent loop statements affect individual tracks in the memory recorder. The other statements affect all tracks.

The first four statements affect pitch:

TRANSPOSE Statement

Just as you can transpose individual tracks on the Synclavier® II, you can transpose individual notelists. Use the word TRANSPOSE followed by a pitch, indicating transpose distance relative to middle C. For example, the statement

```
100 TRANSPOSE G3
```

causes all notes in the notelist to be played a fifth higher than the pitches themselves indicate.

If you want to use a TRANSPOSE statement, you must insert it at the beginning of the notelist, after the NOTELIST statement. There may only be one TRANSPOSE statement in a notelist.

GLIDEFROM Statement

If you have portamento patched into your timbre, the first note in your composition will be preceded by a glide from the lowest C on the keyboard. You can change the starting pitch for the glide with the GLIDEFROM statement. Use the word GLIDEFROM, followed by a pitch. For example, the statement

```
110 GLIDEFROM A2
```

will cause the first note of the notelist to "glide from" the A below middle C.

If you want to use a GLIDEFROM statement, you must insert it at the beginning of the notelist, after the NOTELIST statement. To hear any affect from the GLIDEFROM, you must use a timbre with portamento.

Special Tunings

You can use SCRIPT statements to set the Synclavier® II OCTAVE RATIO button and the SCALE ADJUST button. (See the *Synclavier® II Instruction Manual* for the musical effects of these two controls.)

To set the OCTAVE RATIO, use the word OCTAVERATIO followed by a ratio.

On the Synclavier® II, you make SCALE ADJUST settings in Hertz increments. In SCRIPT you make these settings in terms of the conventional equal temperament scale. Use the word SCALE followed by a list of 12 numbers. Each whole number represents a semitone interval relative to C. Each of these intervals can be tuned in cents (a cent is 1/100th of a semitone).

For example, just tuning for a D major chord (D - F# -A) can be accomplished by lowering F# 10 cents as follows:

```
130 SCALE 0 1 2 3 4 5 5.90 7 8 9.02 10 11
```

The next two statements affect rhythm:

SPEED Statement

You can use the SPEED statement to access the SPEED control in the Synclavier® II memory recorder and speed up or slow down all tempos in your composition. Use the word SPEED, followed by a number. For example, the statement

```
140 SPEED 1.500
```

causes the entire composition to be played at 1.5 times the rate specified in the TEMPO statement. (The default SPEED setting is 1.000 and does not affect tempo at all.)

CLICKPERIOD Statement

You can access the Synclavier® II CLICK RATE button with the CLICKPERIOD statement. Whereas click rate is presented in clicks per minute in the Synclavier® II display window, in SCRIPT you specify the period in seconds that you want to elapse between clicks. Thus, a click rate of 120 clicks per minute is written as

```
150 CLICKPERIOD 0.5
```

The formula for calculating CLICKPERIOD is

```
    CLICKPERIOD = 60 / clicks per minute
```

CLICKPERIOD settings are accurate down to .005 seconds. Setting up the click rate in relation to the tempo makes it easy to synchronize additional sequences generated on the Synclavier® II keyboard with your composition.

LOOP Statements

You can enter overall *Da capo* loops in SCRIPT. The LOOP statement inserted in a composition will access the LOOP button on the Synclavier® II. Use the keyword LOOP followed by the time in seconds at which the sequence should loop back.

Here is the composition KEY with a loop.

```
100 KEYSIG D
110 LOOP 6.000
120 NOTELIST USING 8-4
130
140  P   D  E  F    G  G#  A  G       F
150  R   4  4  2    4  4   4  4       1
160
170 END
180
190 NOTELIST USING 8-4
200
210  P   F  G  A    B  C4  D  E  C     D
220  R   4  4  2    2  8   8  8  8     1
230
240 END
```

If you like, you can play a sequence of SCRIPT notes after the loop point. Press the LOOP button during performance to remove the loop.

You can also enter independent loops or separate loops on each notelist. Examine the following notelist:

```
100 NOTELIST USING 1-1
110 TEMPO 4 = .25 SECOND
120 STARTLOOP HERE
130   P  C3  D  E  F  G  A  B  C
140   R  4      4  4  4  4  4  4  4
150 ENDLOOP HERE
160 END
```

Note that tempo is expressed in terms of seconds on line 110. You must express tempo in seconds or frames when establishing an independent loop. On line 120, the words STARTLOOP HERE are entered. Since they appear before the first note in the composition, a *Da capo* independent loop is established. The words ENDLOOP HERE, entered on line 150, establish the loop point.

You can also create *Dal segno* independent loops, or loops preceded by an introduction. And you can enter additional notes to be played after the loop point. To hear such notes, you must remove the loop during performance by holding down the track button and pressing LOOP at the right moment.

An example follows:

```
100 NOTELIST USING 1-1
110 TEMPO Q = .25 SECOND
120   P  G3  F  E  D
130   R  2      2  2  2
140 STARTLOOP HERE
150   P  C3  D  E  D  C3  D  E  D  C3  B2  C3  D  E  F  E  D
160   R  8      8  8  8  8     8  8  8  8     8     8     8  8  8  8  8
170 ENDLOOP HERE
180   P  C3  B2  C3
190   R  2      2     1
200 END
```

Note that you should make all independent loops an *even* number of beats long. Doing so will ensure that the loop will remain synchronized at all times with musical events taking place on the other tracks.

THE INSERT STATEMENT

Because some compositional elements are useful in more than one composition, we have provided an additional statement: the INSERT statement. You can use this statement to create a single composition from several different files on your user diskette. Use the word INSERT followed by the name of the file you want inserted when the composition is played. The name of the file must be enclosed by quotation marks.

```
100 INSERT 'filename'
```

One can make use of the INSERT statement in a very long composition. Since there is a limit to the length of a file you can edit, you can store sections of such a composition in separate files and link them in one composition with INSERT statements.

There are a variety of other uses for the INSERT statement which will be covered in later chapters.

MORE ABOUT THE MONITOR

Now that you are familiar with the basic operation of the SCRIPT monitor and with SCRIPT musical notation, you are almost ready to enter your own compositions. First, you need to learn how to create new files and store them on diskette, and how to correct errors and otherwise edit a composition.

Compositions in SCRIPT notation consist of lines of typed text that, in contrast to monitor commands, all begin with line numbers. When you finish typing a line and press the RETURN key, the monitor will sort it by line number. In the case of duplicated line numbers, the new line will replace the old one. Any line entered through a terminal that does not begin with a number is interpreted by the monitor to be a command. If an unnumbered line is entered that is not a legitimate command (for example, if you forget the line number), the monitor will respond by writing "WHAT?".

ENTERING A NEW COMPOSITION

For practice in entering a new composition, begin by creating a duplicate of our simple composition of Example 1.

1. Type the command NEW followed by a file name, like MYTUNE, which is different from the file names already in the catalog of your SCRIPT user diskette.

   File names can consist of up to eight letters or numbers. The monitor command NEW allows you to enter a new composition into computer memory.

2. Then, line by numbered line, type the following list:

```
100 NOTELIST USING 8-1
110   P  C  D  E  E  F  G  C
120   R  4  4  2  4  4  2  1
130 END
```

3. Type PLAY
4. If the SCRIPT compiler indicates errors, retype the line in question.
5. After your composition plays, return to the monitor by typing S.
6. Type SAVE

   The monitor command SAVE stores the current file, your composition, MYTUNE, on the user diskette.

7. Type CAT

   You should see the name MYTUNE in the catalog.

   As you store more and more files on your diskette, you will eventually run out of room. You can always use additional diskettes as described in "SCRIPT User Files." But you can also create more room for new files on any diskette by removing those no longer being used.

8. Type UNSAVE MYTUNE

   The monitor command UNSAVE completely erases the designated file from the user diskette.

9. Type CAT

   The file name MYTUNE is gone from the catalog.

(NOTE: When the conversion program translates a Synclavier® II sequence into SCRIPT, it places that translated version in the current file. As that occurs, the original contents of the current file will be erased. Therefore, if you plan to use the conversion program, you should SAVE the composition on diskette *before* typing PLAY.)

CORRECTING ERRORS OR EDITING

The SCRIPT compiler will point out all your spelling and syntax errors. For example, it might inform you that you have not counted pitches and rhythms carefully, or that some SCRIPT word you have entered is misspelled.

If no errors are found by the compiler, the piece will be played by the Synclavier® II performance system. At this time your ear will inform you about errors of pitch, rhythmic value, tempo, etc. An examination of the composition's listing will tell you what needs changing.

Often, the easiest way of correcting errors in SCRIPT is to retype the line with a mistake in it. SCRIPT lines are very short. There are, however, a variety of editing commands available. You can use them to correct errors, move blocks of lines around, to delete a line or lines, or to append one file to another.

For practice in editing, try the following:

1. Type OLD ERRORS
2. Type RENAME CORRECT

   The monitor command RENAME changes the name of the current file. By renaming this file CORRECT, you can practice making corrections without changing the demonstration file itself.
3. Type PLAY

   You notice the composition is not playing. Instead you see the following list of errors on the terminal.

```
UNDEFINED SYMBOL "TENPO" IN LINE 00110
STATEMENT NOT ALLOWED OUTSIDE NOTELIST AT LINE 00110
MISSING OR UNRECOGNIZED TIMBRE NAME AT LINE 00110
RHYTHM LINE DOES NOT MATCH PITCH AT LINE 00160
```

4. Type LIST to see what caused the error messages.

   Sometimes, one error can result in several error messages. The misspelling of TEMPO causes the first two error messages. The third error message is caused by the numbers "8-9" after NOTELIST USING (there is no entry 9 in a timbre bank). The fourth error message is caused by an unequal number of pitches and rhythmic values in lines 150 and 160.
5. Type the lines over to correct them; then type PLAY. The composition should play correctly.

Since you gave the current file a new name, you could type SAVE to store the edited composition on diskette. If you did, you would have two versions: the original, ERRORS, and the edited, CORRECT.

The REPLACE command is an alternative way of storing an edited file. In this case, you do not RENAME the file recalled from diskette. After placing a stored file in the current file, you make the desired changes and then type REPLACE. The edited version in the current file will replace the original version on the diskette. You will have saved only the latest version.

You can also replace a file with a different name than the current file by following the command REPLACE with the different file name.

EDITING ALTERNATIVES

There are additional monitor commands that can make your editing quicker. You can use these commands to change a word or a number instead retyping an entire line. Or to delete groups of lines or to move lines around.

The CHANGE Command

Instead of retyping line 110 in ERRORS, you could have used the monitor command CHANGE as follows:

```
CHANGE TENPO,TEMPO
```

In this command, the word CHANGE is followed by old spelling (comma) new spelling; it causes the first occurrence of TENPO in the file to be replaced by TEMPO. These spellings are called *literal strings.* In literal strings, the case of the letters is not disregarded. If a string is upper case in the numbered line, it must be upper case in your CHANGE command. TENPO is not the same literal string as tenpo.

Additional Uses of the CHANGE Command

If there were two TEMPO statements in the composition and you misspelled them both TENPO, you would type

```
CHANGE TENPO,TEMPO, ALL
```

This command would change all occurrences of TENPO in the composition to TEMPO.

You might want to change all the C sharps in your composition to C naturals. To do so, type

```
CHANGE C#,CN, ALL
```

You can also specify a line number after the new spelling to designate where you want the change to occur.

```
CHANGE C#,CN, 110
```

In this example, the CHANGE command would change only the C sharp in line 110 to C natural.

In addition, you can change numbers as well as letters.

```
CHANGE 90,75, 150
```

In this example, the CHANGE command would change the 90 in line 150 to 75.

When you start editing Synclavier® II sequences that have been converted to SCRIPT (or SCRIPT compositions in the electronic notelist format), you will find the CHANGE command very useful.

The EXTRACT Command

The monitor command EXTRACT can be used to extract, or preserve, a line or lines from the current file and delete all others. Type the word EXTRACT, followed by the number or numbers of the lines you wish to extract.

```
EXTRACT 15
```

In this example, the EXTRACT command would delete all lines but 15.

```
EXTRACT 10, 20–30, 100–150
```

In this example, the EXTRACT command would delete all lines but 10, 20–30, and 100–150.

NOTE: When using this command, you must enter the numbers in increasing order.

The DELETE Command

The monitor command DELETE can be used to erase a line or lines from the current file. Type the word DELETE, followed by the number or numbers of the lines you want to erase.

```
DELETE 20
```

In this example, the DELETE command would erase line 20.

```
DELETE 20, 40–45
```

In this example, the DELETE command would erase lines 20 and 40–45.

The RESEQUENCE Command

The RESEQUENCE command is available for changing line numbers. When you type RESEQUENCE, all the lines in your current file will be renumbered. The first line will become 100 and each line number will be ten greater than the preceding number. Use this command when you want to enter a new line between two consecutively numbered lines, or when you want to display a "clean" list.

The APPEND Command

You can use the monitor command APPEND to join a file stored on the diskette onto the end of the current file. Use the word APPEND, followed by the file name or names of the files to be appended. The file names are separated by commas. The command also causes the lines of the current file to be resequenced starting with 100.

```
APPEND POLLY2
```

In this example, a file named POLLY2 would be added on to the current file and the result would be renumbered.

```
APPEND POLLY2, POLLY2
```

In this example, POLLY2 would be appended twice.

The MOVE Command

You can use the monitor command MOVE to move a line or a block of lines to a new location in the composition. Use the word MOVE, followed by the number or numbers of the lines you wish to move, a comma, and then a single line number which designates the new location. After the MOVE is made, the file is resequenced automatically.

```
MOVE 10, 110
```

In this example, the MOVE commmand would place line 10 after line 110 and renumber the file.

```
MOVE 10–20, 110
```

In this example, the MOVE command would remove lines 10–20 from their current location, reinsert them after 110, and then renumber the file.

The LOCATE Command

The monitor command LOCATE lets you search for a literal string either on one line, a block of lines, or throughout the composition as follows:

```
LOCATE C3
```

In this example, the first occurrence of C sharp would be listed on your terminal.

```
LOCATE C#, 10, 20–50
```

In this example, the occurrences of C sharp in line 10 and in lines 20–50 would be located and listed on your terminal in 19-line segments as in any listing.

```
LOCATE C#, ALL
```

This command would produce a list of all occurrences of C sharp in your composition.

The BUILD Command

The monitor command BUILD is a useful one. If you type BUILD, the computer will automatically type line numbers for you to use when entering text. The numbers start with 100 and increase by 10's. You can use the BUILD mode when beginning a new composition or when adding to the end of the current file.

To leave the BUILD mode, press RETURN without entering any text after a line number appears.

Additional monitor commands are listed in the SCRIPT Reference Manual. NOTE: The SCRIPT monitor and editor is very similar to the Scientific XPL monitor described in detail in the New England Digital publication, *Creating Programs for ABLE Series Computers.*

You can control every aspect of the Synclavier® II through the SCRIPT composition language. In the previous examples, we have used SCRIPT musical notation to control the Synclavier® II memory recorder. For timbres, we have used those previously created on the Synclavier® II and stored on diskette in the timbre banks. You can also define timbres in SCRIPT, using the SCRIPT timbre definition format. This is also the format into which the SCRIPT conversion program translates the timbres from the Synclavier® II memory recorder.

This chapter provides a brief introduction to SCRIPT timbre definitions. More detailed information may be found in the SCRIPT Reference Manual.

A SCRIPT timbre definition must be organized in a definite order. Partial timbres are defined first. Then, complete timbres are assembled which include the partial timbres and any other special effects. Symbolic names, chosen by the user, are used to identify partial timbres and timbres.

You use a SCRIPT timbre definition in a composition by following the word NOTELIST with a *timbre symbolic name* (described below) instead of the USING expression.

A SCRIPT timbre definition can be entered line by line at the beginning of a composition. But it will be available only to the notelists in that one composition. Another method is to store the timbre definitions separately in their own file. Then, by using an INSERT statement, you can use the timbre definitions in any new composition.

The file TIMBRES on your user diskette contains an example of a sound created in a SCRIPT timbre definition. This timbre is used in the BACH13 composition on your user diskette.

```
100 define partial flute1       /* a simple partial timbre */
110
120    volenv 0 10 500 500 100 50
130    harmonics 100 50 33 25 12.5
140    tuning 440.0
150    vibrato sine 6.50 .10 500
160    decay .100
170 end flute1
180
190 define timbre flute
200    partial flute1
210 end flute
```

```
100 define partial flute1
```

You begin a SCRIPT timbre definition by establishing the settings for all partial timbres to be used in the timbre. (If you have used the Synclavier® II, you are familiar with partial timbres. See the *Synclavier® II Instruction Manual* for the musical effects of different values in partial timbre definitions.)

The words DEFINE PARTIAL begin the definition of a partial timbre. A user-supplied symbolic name for the partial timbre follows, in this case, FLUTE1. A symbolic name may be any length but should include only letters, digits, or the special character "." (period). It should include no other punctuation or special marks, such as quotation marks or dashes. Other possible symbolic names might be A1, XYLOPHONE3, or KRUMHORN.

(Note: You cannot use the name of a SCRIPT statement for a symbolic name.)

A line with the word END followed by the same partial timbre symbolic name completes the partial timbre definition. Between the DEFINE PARTIAL line and the END line are all the settings which define the sound of the partial timbre. The settings are entered in eight possible statements. If any of the statements are not entered, default settings are automatically used by the SCRIPT system. (See the SCRIPT Reference Manual for more details about the partial timbre statements.)

In partial timbre FLUTE1, five partial timbre settings are specified. Each statement begins with a SCRIPT word, followed by a number indicating the setting (or what you would see in the digital display window on the Synclavier® II). The numbers are separated by spaces.

```
210 volenv 1 10 500 500 100 50
```

The VOLENV statement establishes the volume envelope settings. The first four numbers refer to DELAY, ATTACK, INITIAL DECAY and FINAL DECAY time in seconds; the fifth and sixth, to PEAK and SUSTAIN level. The numbers are in the same order as the volume envelope buttons on the Synclavier® II control panel.

```
220 harmonics 100 50 33 25 12.5
```

The HARMONICS statement establishes the harmonic coefficient settings. Up to 24 numbers can be listed. If less than 24 are listed, the rest of the harmonics will be set to zero. More than one line may be used.

```
230 tuning 440.0
```

The TUNING statement establishes the PARTIAL TUNING setting for the A above middle C in Hertz.

```
240 vibrato sine 6.50 .10 500
```

The VIBRATO statement establishes the four VIBRATO settings. The first is WAVE shape (SINE), the second is RATE in Hertz, the third is DEPTH in semitones, and the fourth, ATTACK time in milliseconds.

```
250 decay .100
```

The DECAY statement establishes the DECAY ADJUST setting.

```
260 end flute1
```

As stated above, the partial timbre is terminated by the word END followed by the symbolic name (spelled exactly the same as when first entered).

270

280 define timbre flute

The words DEFINE TIMBRE begin the complete timbre definition, in this case the simplest and shortest one possible since only one partial timbre and no special features have been used. Note that the timbre symbolic name must be *different* from the partial timbre symbolic name FLUTE1. The rules for timbre symbolic names, however, are the same as those for partial timbre symbolic names.

290 partial flute1

The symbolic name for each partial timbre to be included is preceded by the word PARTIAL. This simple timbre definition lists only one partial. Up to four partial timbres can be included in a timbre definition. More complicated timbre definitions, involving more than one partial and CHORUS, REPEAT, and ARPEGGIATE functions and REAL-TIME EFFECTS PATCHING are covered in the SCRIPT Reference Manual.

300 end flute

The timbre definition ends in the same way as partial timbres, with END and the symbolic name.

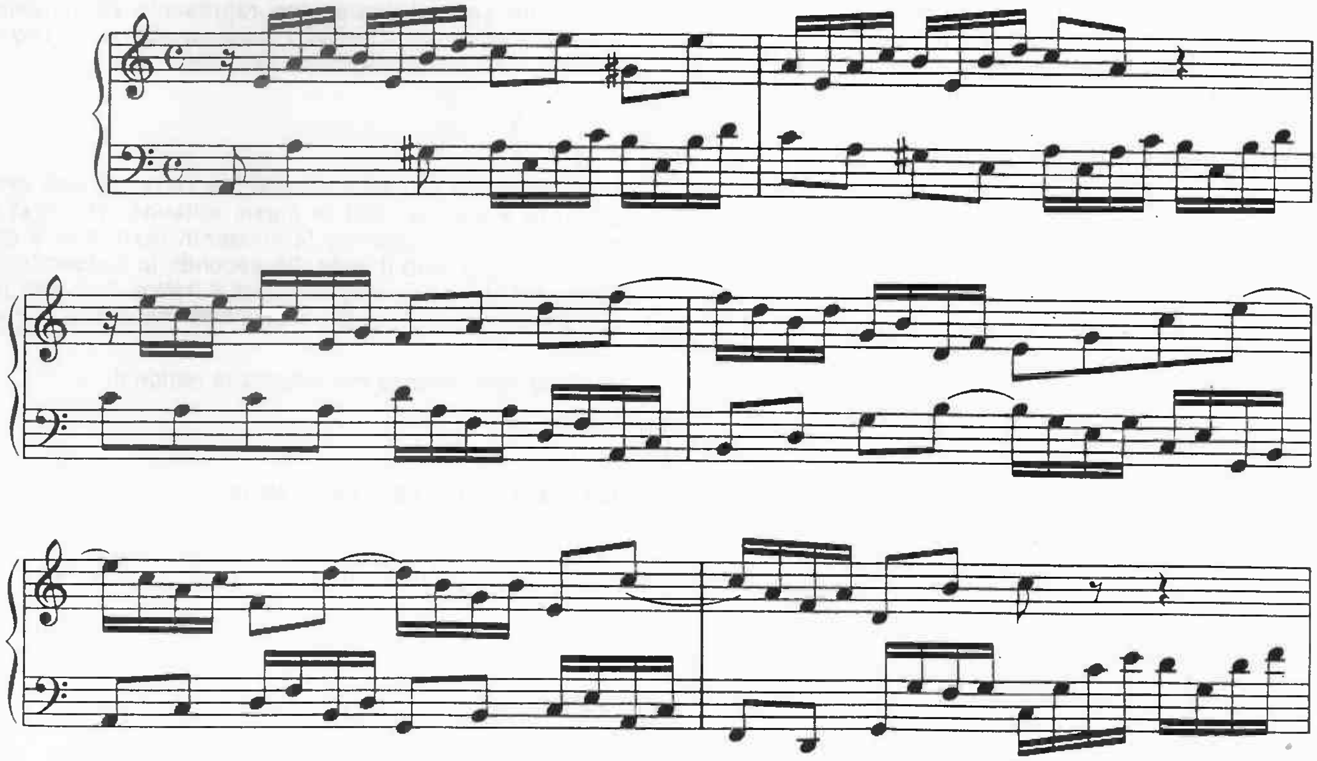
The timbre is now defined and may be used in a notelist.

In SCRIPT musical notation, notes are entered in the notelist in terms of beats. For composers of computer music, we have provided a second notelist format, one that is based on time (specified in seconds or in frames). This is also the format into which the SCRIPT reverse compiler translates Synclavier® II memory recorder sequences.

The SCRIPT computer music notelist is based on a format widely used in computer music (it is quite similar to Music V*, for example) and consists of a sequential list of notes, each one on a separate line. The starting time, pitch, duration, and volume and timbre modification percentages of each note are entered on one line.

The simplicity of this format permits exchange between compositions written in SCRIPT and other computer systems; it also makes possible Synclavier® II performance of compositions written on other computer systems.

To demonstrate this format, we have prepared a composition based on the first six bars of the Bach Invention 13 (in two parts) and stored it on the user diskette under the file name BACH13.



* For a description of Music V, see Max V. Mathews, *The Technology of Computer Music* (M.I.T. Press, Cambridge, Mass., 1969).

Below is the beginning of the listing in SCRIPT computer music format. The complete listing can be found in the Appendix or listed on your terminal.

```
100 insert 'timbres'
110
120 notelist flute
130
140  .25   e3   .25
150  .50   a3   .25
160  .75   c4   .25
```

100 insert 'timbres'

The INSERT statement causes the file TIMBRES to be inserted at line 100 when you type PLAY. The file TIMBRES contains the SCRIPT timbre definition which will be used in the BACH13 composition.

120 notelist flute

The first notelist is the treble part of the Invention. The word NOTELIST is followed by the symbolic name FLUTE, a timbre defined in the file TIMBRES.

Within a notelist, the notes must be sorted in the time order that they appear in the composition. Each note is on a separate line and is represented by a series of numbers.

The first number represents length of time, in seconds, from the start of the composition to the beginning of the note. The second expression represents pitch (using a standard pitch as in SCRIPT musical notation). The next number indicates duration in seconds.

```
140  .25   e3   .25
150  .50   a3   .25
160  .75   c4   .25
```

In creating the BACH13 notelist, we have set one second equal to a quarter note in music notation. The first note occurs after .25 seconds (a sixteenth rest); it is E above middle C (E3); and it lasts .25 seconds (a sixteenth note). Since the the beginning time and duration time add up to the beginning time of the next note, the piece is played legato.

An END statement to the notelist is optional.

970 notelist flute

990   0.0   a1   .50

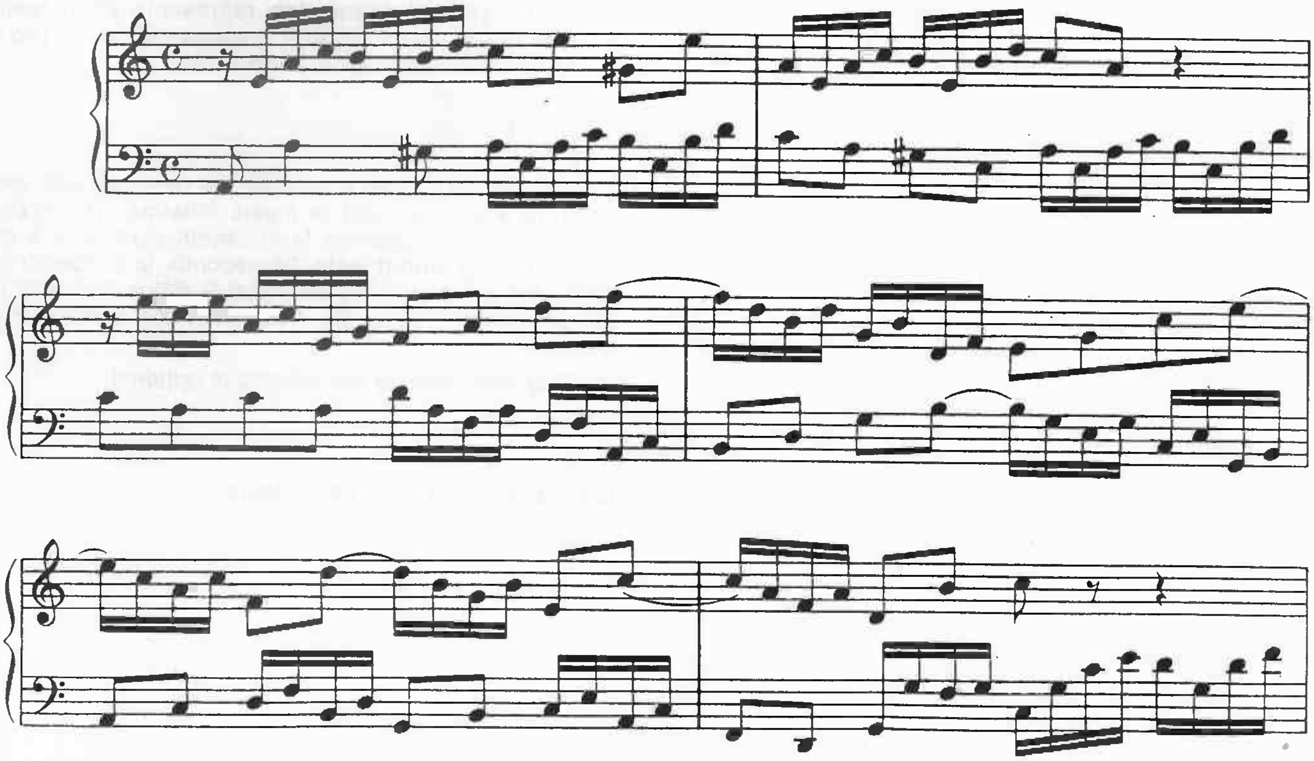
On line 970, the bass part begins.

In SCRIPT musical notation, notes are entered in the notelist in terms of beats. For composers of computer music, we have provided a second notelist format, one that is based on time (specified in seconds or in frames). This is also the format into which the SCRIPT reverse compiler translates Synclavier® II memory recorder sequences.

The SCRIPT computer music notelist is based on a format widely used in computer music (it is quite similar to Music V*, for example) and consists of a sequential list of notes, each one on a separate line. The starting time, pitch, duration, and volume and timbre modification percentages of each note are entered on one line.

The simplicity of this format permits exchange between compositions written in SCRIPT and other computer systems; it also makes possible Synclavier® II performance of compositions written on other computer systems.

To demonstrate this format, we have prepared a composition based on the first six bars of the Bach Invention 13 (in two parts) and stored it on the user diskette under the file name BACH13.



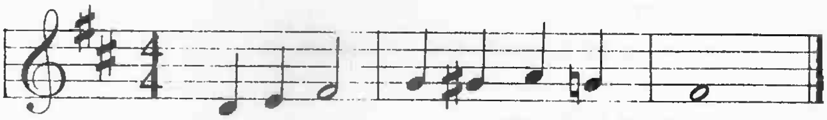
* For a description of Music V, see Max V. Mathews, *The Technology of Computer Music* (M.I.T. Press, Cambridge, Mass., 1969).

More on Starting Times

There are two kinds of starting times for the notes: absolute and relative. The starting times of the notes in the example above are absolute and represent the length of time from the start of the composition. A notelist with absolute starting times is compatible with Music V. And, it may be a convenient format when notes must be synchronized with other events, as in film and video. You may edit out notes without affecting the starting times of subsequent notes (you are creating a rest) and without affecting the chords formed by subsequent notes with notes in other notelists. Moreover, when using the absolute format, the timbre may be changed in the middle of a notelist just by inserting another NOTELIST statement.

You can also use relative starting times. In this case, the starting time of each note is relative to the starting time of the previous note. This form makes it easy to edit out a single note or group of notes without introducing a rest. Simply enter a plus sign before a starting time to render it relative rather than absolute.

The following example shows the same passage in standard notation with absolute and relative starting times:



Absolute Starting Times

```
800 Notelist  Timbre1
810    0.000  D3     0.500
820    0.500  E3     0.500
830    1.000  F#3    1.000
840    2.000  G3     0.500
850    2.500  G#3    0.500
860    3.000  A3     0.500
870    3.500  G3     0.500
880    4.000  F#3    2.000
```

Relative Starting Times

```
800 Notelist  Timbre1
810+  0.000  D3     0.500
820+  0.500  E3     0.500
830+  0.500  F#3    1.000
840+  1.000  G3     0.500
850+  0.500  G#3    0.500
860+  0.500  A3     0.500
870+  0.500  G3     0.500
880+  0.500  F#3    2.000
```

Time in Frames

Starting times *and* note durations can be expressed in frames. You can use the SCRIPT default frame rate of 24 frames per second or establish a different frame rate with an FPS (frames per second) statement (see page 14) before the notelist.

The format for specifying time in frames is:

```
        :hour:minute:second:frame
```

Frames are numbered starting with :0.00. They can be specified to an accuracy of .01 of a frame. Each specification must begin with a colon (":") to inform the SCRIPT compiler that times are specified in frames (and not seconds).

The following note specified in frames

```
100    :1:35:39:12  C          :1:12
```

begins at 1 hour, 3 minutes, 39 seconds and 12 frames after the start of the composition and lasts 1 second and 12 frames. The hour, minute, and second specifications are optional. Another starting time could be entered as

```
        :13:10:12
```

indicating 13 minutes, 10 seconds and 12 frames, or

```
        :39:12
```

indicating 39 seconds and 12 frames, or

```
        :12
```

indicating 12 frames.

When you specify a note just in frames, you can specify any frame number up to 29,999.

Below is our music notation example specified in frames:

```
100 FPS 24
110 Notelist            Timbre1
120             :00.0   D3        :12.0
130             :12.0   E3        :12.0
140          :01:00.0   F#3    :01:00.0
150          :02:00.0   G3        :12.0
160          :02:12.0   G#3       :12.0
170          :03:00.0   A3        :12.0
180          :03:12.0   G3        :12.0
190          :04:00.0   F#3    :02:00.0
```

Note that the first note starts on frame 00.0 on line 120.

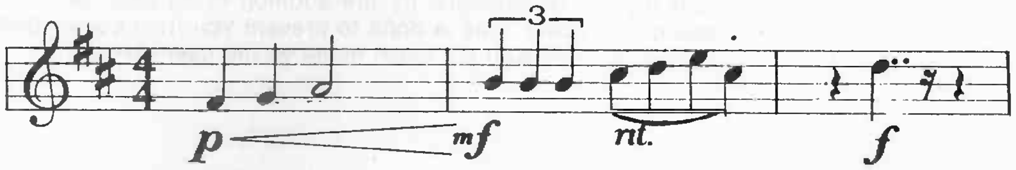
Relative starting times in frames (using the plus sign) are also allowed.

Specifying Volume and Timbre Changes

A volume and a timbre modification percentage may be entered on each line. The volume percentage follows the duration of the note. The timbre modification may be entered next and is enclosed in parentheses.

You can end the line with the S symbol, which causes the note to be slurred with the previous note. For slurring to be heard, however, the starting time plus duration of the previous note must overlap the starting time of the slurred note.

The following example shows the same passage in standard notation listed with volume and timbre specification and slurs:



```
100 Notelist  using 8-5
110  0.000   F#3    0.285   20   (50)
120  0.475   G3     0.285   25   (60)
130  0.950   A3     0.575   30   (75)
140  1.905   B3     0.095   40   (75)
150  2.220   B3     0.095   45   (75)
160  2.540   B3     0.095   50   (75)
170  2.855   C#4    0.340   70   (75)
180  3.095   D4     0.370   74   (75)   S
190  3.365   E4     0.320   80   (75)   S
200  3.685   C#4    0.095   60   (100)
210  4.540   D4     1.145   80   (100)
```

Additional Statements

You may use any of the Synclavier® II control statements or conducting statements available in SCRIPT music notation in a SCRIPT computer music notelist with a few exceptions: Because the TEMPO statement interprets notes in terms of beats; it cannot be used to modify notes entered in terms of time. Furthermore, the duration of notes cannot be set by the ARTICULATION statement.

The SCRIPT reverse compiler can quickly convert a Synclavier® II sequence in the memory recorder into the computer music format for display and editing on the terminal. This output can take several different formats, each of which is designed for convenience in specific situations. The starting times of the notes can be expressed in either absolute or relative time; both the starting times and durations of the notes can be expressed in seconds or in frames.

You can activate the reverse compiler at any time during real-time Synclavier® II operation by pressing certain keys on the terminal. Whenever you do, Synclavier® II operation will stop, and the monitor will be restored; the converted sequence will replace the current file. At the same time the name of the current file will be changed by the addition of an asterisk to the end of the name. This is done to prevent you from accidentally *replacing* a file with the same name on the user diskette.

ABSOLUTE OR RELATIVE

The two formats for note starting times, absolute and relative, are designed for different kinds of editing.

Absolute starting times are useful when editing polyphonic compositions. An incorrect or superfluous note on one notelist can be edited out by deleting or changing the appropriate line of text without affecting the timing or harmonization of succeeding notes. Similarly, notes can be added without affecting subsequent notes.

You can also change a timbre in the middle of a sequence of notes with absolute starting times. Simply insert an additional NOTELIST USING statement before the line containing the first note you wish to be played in the new timbre.

On the other hand, if you select a format with relative note starting times, you can remove a note or an entire measure without leaving any rests. Subsequent notes will automatically move forward in time. You may have to adjust the starting time of the first note following the removed notes so that it will come out on the right beat. All other note starting times will be adjusted for you.

Relative note starting times are also useful when you want to move entire musical passages from one location to another or when you want to duplicate a passage.

TIMES IN FRAMES

The time formats which express starting times and durations in frames are especially useful for film and video synchronization. You can choose from frame rates of 24 or 30 frames per second, or, for synchronization with European and other unusual video, choose the "arbitrary" option and specify any frame rate from 1.00 to 300.00 frames per second. The use of the reverse compiler can be combined with the use of SCRIPT synchronization statements to adjust automatically the starting times and tempos of a keybord performance to synchronize it with film or video cues; see the next chapter.

SELECTING A NOTELIST FORMAT

To select a particular format, press the ESC key followed by one of the number keys, 1 through 9. Each number key activates a different format. On a VT100, you may alternately press one of the PF keys on the right side of the keyboard. Pressing PF1 is identical to pressing ESC and 1; pressing PF2 is identical to pressing ESC and 2. Note that the conversions for ESC and 3 and ESC and 4 (or PF3 and PF4) have been reserved for future additions to SCRIPT.

Find the right key in the following table:

	ABSOLUTE TIMES	RELATIVE TIMES
SECONDS	PF1 ESC 1	PF2 ESC 2
24 FPS	ESC 5	ESC 6
30 FPS	ESC 7	ESC 8
ARBITRARY	ESC 9	ESC 9

You will find the reverse compiler useful in other ways. For example, you can extract timbre definitions and store them in their own files on the user diskette. Or you can use the reverse compiler to divide a sequence into several different files.

SEQUENCES TOO LONG TO BE CONVERTED

Notes are stored more efficiently in the Synclavier® II operating system than in text files. A maximum length sequence in a 10,000 note Synclavier® II system will be too long to convert directly into text. If you want to convert such a sequence, follow this procedure:

1. First store your complete sequence on a Synclavier® II sequence diskette.
2. Next, using the ERASE button on the Synclavier® II, erase some of the tracks in the memory recorder.
3. Try conversion again. If not successful, erase more tracks. If successful, SAVE the converted, nonerased notelists.
4. Return to the Synclavier® II, recall the sequence, and erase those tracks that were converted, and convert those tracks that were erased.
5. When all tracks are converted and saved, create a SCRIPT composition by linking all the saved files with INSERT statements.

With SCRIPT music notation, you can easily compose melodic motifs which can then be repeated in one or more notelists, transposed and inverted in both the tonal and strict manner, and played in retrograde.

The use of motifs involves two stages: first, the definition and next, the performance.

DEFINING A MOTIF

A SCRIPT motif definition includes of all the notes of the motif. The first line of the definition begins with the words DEFINE MOTIF followed by a motif symbolic name. This symbolic name may be any length but it must start with a letter. (NOTE: You cannot use the name of a SCRIPT statement for a symbolic name.)

Following this opening statement, you may include a KEYSIG statement which will establish a key signature for the pitches in the motif. This KEYSIG statement will also define the key in which tonal transpositions and tonal inversions will be performed.

If you do not include a KEYSIG statement, then the key signature in effect at the beginning of the motif definition will be used for the motif. If no KEYSIG statement occurs either before or within the definition, then the default key of C will be used.

Any KEYSIG statement which occurs *within* a motif definition will not affect any other notes or notelists occurring after the end of the motif.

Next in the motif definition must appear all the pitches and rhythmic values of the notes in the motif. You enter them just as you would in a notelist, with a pitch line first and then a rhythm line.

For lengthy motifs, you may enter multiple pitch and rhythm lines. In this case, however, all the pitch lines must be grouped together, followed by all the rhythm lines.

You may also include articulation, timbre modification and volume percentage lines in your motif definition.

The last line of the motif consists of the word END, followed by a repetition of the motif symbolic name.

An example of a SCRIPT motif definition follows:

```
100 DEFINE MOTIF MELODY
110 KEYSIG G
120   P  B3   F4   E    A3  A  D  G  F  B  A  G
130   R  4.   8+8  8+8  8   4  8  8  4  4  2  2
140 END MELODY
```

A motif must be defined before the beginning of the notelist in which it is to be cited. It may also be defined and stored in its own user file. You would then enter an INSERT statement with that file name before the notelist in which you wish to use the motif.

PERFORMING A MOTIF

To use the motif, you include a PERFORM statement in the notelist wherever you wish the motif to appear. When the composition is played, the notes of the motif will be performed in the timbre and currently active tempo of that notelist. Any articulation, timbre modification, or volume defaults or multipliers in effect will also be applied to the notes of the motif.

The key signature currently in effect in the notelist does not affect the motif.

The PERFORM statement consists of the word PERFORM followed by the motif symbolic name. For example, the statement

```
200 PERFORM MELODY
```

would perform the notes of the motif defined above at line 200.

TRANSFORMING A MOTIF

There are seven different SCRIPT operations that you may apply to the motif: tonal transpositions, strict transpositions, tonal inversions, strict inversions, pitch only retrogrades, rhythm only retrogrades, and pitch and rhythm retrogrades. List one or more of the following symbols within parentheses after the motif name in the PERFORM statement:

T	tonal transposition
ST	strict transposition
I	tonal inversion
SI	strict inversion
R	retrograde
PR	pitch only retrograde
RR	rhythm retrograde

For example, the statement

```
200 PERFORM MELODY (I)
```

would perform a tonal inversion on the pitches of the motif MELODY; the inverted motif would be inserted into the notelist at line 200.

When using either kind of transposition, you must also indicate the new starting pitch for the motif in parentheses after the T or ST. This starting pitch is affected by the key signature currently in effect in the notelist.

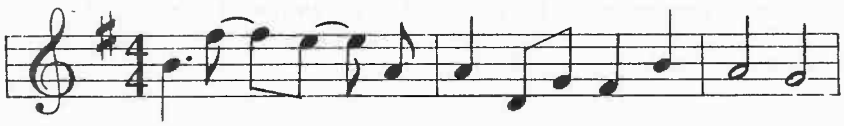
AN EXAMPLE

To demonstrate the use of motifs, we have prepared a short composition which defines a motif, performs it, and then applies the seven operations.

```
100 DEFINE MOTIF MELODY
110 KEYSIG G
120   P  B3    F4   E     A3  A  D  G  F  B  A  G
130   R  4 .   8+8  8+8   8   4  8  8  4  4  2  2
140 END MELODY
150
160 TEMPO 2 = 120
170 NOTELIST USING 8-4
180 PERFORM MELODY
190 P   r
200 R   1
210 PERFORM MELODY (T(D))
220 P   r
230 R   1
240 PERFORM MELODY (ST(D))
250 P   r
260 R   1
270 PERFORM MELODY (I)
280 P   r
290 R   1
300 PERFORM MELODY (SI)
310 P   r
320 R   1
330 PERFORM MELODY (R)
340 P   r
350 R   1
360 PERFORM MELODY (PR)
370 P   r
380 R   1
390 PERFORM MELODY (RR)
400 END
```

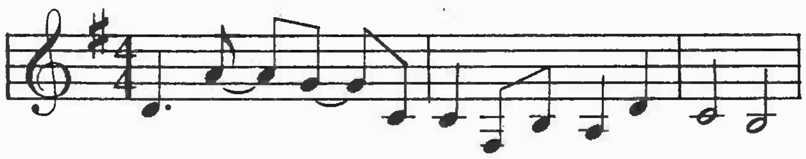
180 PERFORM MELODY

This statement causes the motif to be played as written in its own key signature (G):



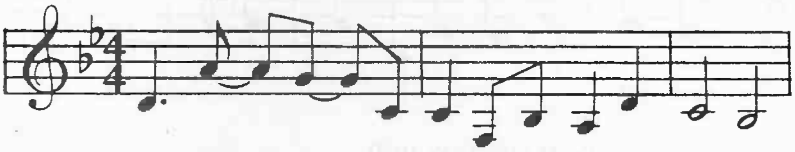
210 PERFORM MELODY (T(D))

This statement causes the motif to be transposed tonally. The transposed motif will begin on the pitch specified in the parentheses after the T, but the motif key signature will remain unchanged. In our example, the transposed motif will begin on D and the key signature of G will be maintained:



240 PERFORM MELODY (ST(D))

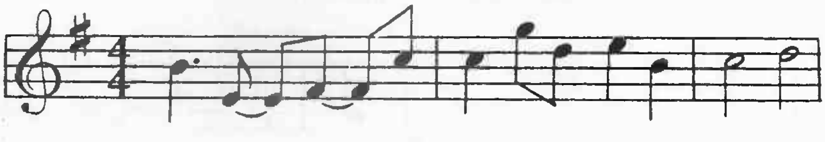
This statement causes the motif to be transposed in the strict manner. As with tonal transposition, the transposed motif will start on the pitch specified in the parentheses after the T, but the exact melodic intervals of the motif will be maintained instead of the key signature. Hence, the motif will be transposed to the key of B flat:



When you use the Synclavier® II's TRANSPOSE button and when you insert TRANSPOSE statements in a note-list, you are applying strict transpositions on either all the notes in the memory recorder or all the notes on one track. In both cases, you can remove the transpositions from the notes during real-time performance by using the TRANSPOSE button. Transpositions on motifs cannot be removed in that way, because they affect only some of the notes on a track.

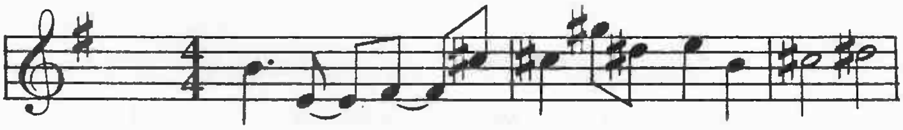
270 PERFORM MELODY (I)

This statement causes the motif to be inverted tonally around its starting pitch, while maintaining the key signature of the original motif:



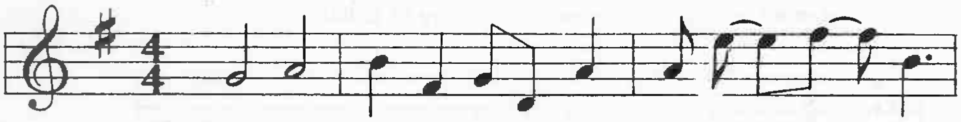
300 PERFORM MELODY (SI)

This statement causes the motif to be inverted in the strict manner around the starting pitch without regard to the key signature. The exact melodic intervals of the motif will be maintained except they will be inverted in the opposite direction. In other words, both original and inverted intervals will contain exactly the same number of semitones:



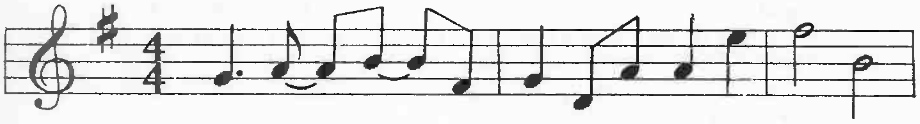
330 PERFORM MELODY (R)

This statement causes the motif to be performed in retrograde. The motif will be played beginning with the last note and ending with the first, without any pitch or duration modifications:



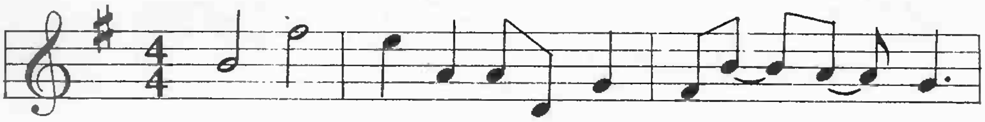
360 PERFORM MELODY (PR)

This statement causes only the pitches of the motif to be performed in retrograde; the rhythmic values will be performed in a forward manner:



390 PERFORM MELODY (RR)

This statement causes only the rhythmic values of the motif to be performed in retrograde; the pitches will be performed in a forward manner:



ACCIDENTALS IN TONAL TRANSPOSITIONS

You may have wondered what happens in tonal transposition when your motif starts on a pitch that is an accidental in the motif key signature or when you attempt to transpose to a starting pitch that is an accidental in the motif key signature.

In either case, the SCRIPT compiler will modify the pitches of the notes in the following manner: First it will transpose the motif tonally to the appropriate naturally occurring pitch; then it will raise or lower all pitches in the motif additional semitones until the starting pitch matches the pitch specified in the PERFORM statement.

For purposes of this discussion we have defined a natural as any pitch naturally occurring within the key signature of the motif. For example, C# would be natural in a motif key of D. An accidental is a pitch which would not naturally occur within the key signature of the motif.

We have prepared a second demonstration composition which includes these four tonal transpositions: 1. transposing a natural pitch to another natural pitch; 2. transposing an accidental pitch to a natural pitch; 3. transposing a natural pitch to an accidental and 4. transposing an accidental to an accidental.

The composition follows:

```
100 DEFINE MOTIF ONE
110 KEYSIG C
120   P  C#3  D  E  D  C
130   R  4    4  4  4  4
140 END ONE
150
160 DEFINE MOTIF TWO
170 KEYSIG C
180   P  C3  D  E  D  C
190   R  4   4  4  4  4
200 END TWO
210
220 NOTELIST USING 8-4
230 PERFORM TWO (T(F))
240 PERFORM ONE (T(F))
250 KEYSIG D
260 PERFORM TWO (T(F))
270 PERFORM ONE (T(F))
280 END
```

Both motifs are in the key of C. Motif ONE begins with an accidental (C#). Motif TWO begins with a natural pitch within the motif key of C.

Natural Tonal Transposition

Line 230 causes a normal tonal transposition of motif TWO from C to F. The transposed pitches would be:

F G A G F

Accidental Transposed to Natural Pitch

Line 240 causes motif ONE, which starts with an accidental, to be transposed tonally to F, which is a natural within the motif's key of C. The transposed pitches would be:

F Gf Af Gf Ff

Natural Pitch Transposed to an Accidental

Line 260 causes motif TWO, which starts with a natural pitch in the motif's key, to be transposed tonally to F#, which is an accidental in the motif's key. The transposed pitches would be:

F# G# A# G# F#

Accidental Transposed to an Accidental

Line 270 causes motif ONE, which starts with an accidental, to be transposed tonally to F#, which is also an accidental. The transposed pitches would be:

F# G A G F

ACCIDENTALS IN TONAL INVERSIONS

Accidentals in the motif also affect the performance of tonal inversions. If the starting pitch is an accidental, the opposite accidental will be applied to it during inversion. Then, all other notes will be inverted tonally around this new pitch retaining the sharps and flats of the motif's key signature. If any notes besides the starting note are accidentals, they will have the opposite accidental applied during inversion.

COMBINING OPERATIONS

You can specify more than one transformation operation in the same PERFORM statement. Tonal inversions may be combined with strict inversions, tonal transpositions, and strict transpositions to create almost any transformation of a motif.

To combine operations, simply list additional symbols in the parentheses. The operations will be performed in the order in which they are listed.

For example, the statement

```
100 PERFORM MELODY (I R)
```

would perform a tonal inversion on the motif and then retrograde it. The statement

```
110 PERFORM MELODY (I T(E3))
```

would perform a tonal inversion on the motif and then transpose it. The inverted motif would begin on E3.

If you follow a strict transposition with a tonal transposition or a tonal inversion, the tonal operations will be performed in relationship to the key signature of the transposed motif, rather than the key signature of the original motif.

Special key signatures using double sharps or double flats are automatically created as required by the SCRIPT system.

USING RETROGRADES

Most motifs can be performed in retrograde with no problems, even motifs with braces and brackets around the pitches. The slurred pitches produced by braces will be played backwards correctly. Bracketed pitches with the same rhythmic values can be retrograded correctly as well. However, if the rhythmic values associated with the bracketed pitches are different, somewhat unexpected results will occur.

Pitch only or rhythm only retrogrades performed on a motif with bracketed pitches can also produce unexpected (though logical!) results.

CHANGING ARTICULATION, TIMBRE MODIFICATION AND VOLUME PERCENTAGES

As you have seen, articulation, timbre modification and volume lines can be included in a motif definition. They can also be included immediately after the PERFORM statement. If you place them there, they will override the percentages specified in the motif definition.

Examine this motif definition:

```
100 DEFINE MOTIF MELODY
110 P  C3   D    E    F    D    E    C
120 R  4    4    4    4    4    4    2
130 V  30   40   50   55   50   35   25
140 END MELODY
```

and this PERFORM statement:

```
170 PERFORM MELODY
180 V  30  40  50  55  65  85  100
```

The volume percentages in line 180 will be used instead of those in line 130.

This chapter describes the SCRIPT statements which accomplish tempo changes on all notelists simultaneously. The first group of statements simplify entering accelerandos and ritardandos on all tracks in synchrony. The second group make possible the synchronization of SCRIPT compositions or real-time performances with time cues expressed in frames or in seconds.

TEMPO CONTROL STATEMENTS

Overall tempo control statements are used with SCRIPT music notation to establish the tempos for all notelists for the entire length of the composition. When you use this form of tempo control, you cannot use TEMPO statements within notelists, as described in the chapter on "The SCRIPT Language".

The overall tempo control statements are grouped together in one section before all notelists and before the synchronization statements (described in the next section).

The first statement in the tempo control section is a TEMPO statement which will establish an initial tempo on all the notelists. You may establish an initial tempo based on beats per minute, seconds, or frames. (See section on the TEMPO statement in "The SCRIPT Language".) Only this one TEMPO statement may appear in the composition.

This statement is followed by a series of "AT" statements which establish the tempo changes and when they are to occur. They also determine whether the changes in tempo are to occur abruptly or gradually.

For an abrupt tempo change, use an AT statement in this format:

```
AT BEAT beat number TEMPO new tempo
```

The first beat in the composition is numbered 1. The beat number is counted in the rhythmic values established in the previous TEMPO and AT statements. The new tempo is specified in beats per minute or by period (seconds or in frames).

The words ACCEL TO and RITARD TO can be substituted for the word TEMPO in this AT statement; the different words are provided for your convenience.

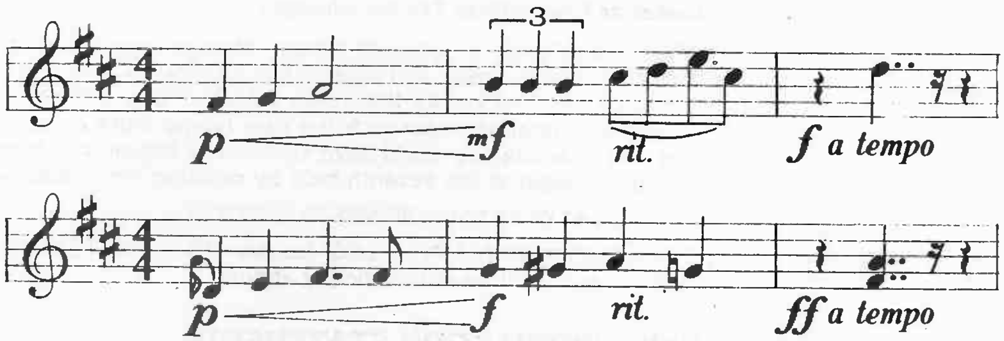
For a gradual tempo change, use an AT statement in either of these formats:

```
AT BEAT beat number ACCEL TO new tempo IN beats
or
AT BEAT beat number RITARD TO new tempo IN beats
```

The word IN is followed by the number of beats you want the tempo change to be extended over. You cannot use the word TEMPO in a statement with the IN expression, but you may use either ACCEL TO or RITARD TO.

An Example

In an earlier example (DYNAMICS), we used a rather clumsy method to establish a synchronized ritardando on two notelists. Now we will show you how to use overall tempo control statements to accomplish the same thing with much less effort. Here is the standard notation again:



Here is the SCRIPT notation using the overall tempo control statements. This composition is on your user diskette under the file name TEMPO.

```
100 TEMPO 4=126
110 AT BEAT 7 RITARD TO 4=72 IN 2
120 AT BEAT 9 TEMPO 4=126
130
140 KEYSIG D
150 TIMBRE 75
160
170 NOTELIST USING 8-5
180
190  P  F   G   A   B   B   B   {C4  D    E}   C   R  D
200  R  4   4   2   (4T)3        8   8    8    8   4  7/16
210  V  20  25  30  40  45  50  70   75   80   60  0  80
220  A  60  60  60  30  30  30  100  100  100  25  0  100
230
240 END
250
260 NOTELIST USING 8-5
270
280  P  A3   D     E   F   F   G   G#  A   G   R  [  F     A  ]
290  R  32   4-32  4   4.  8   4   4   4   4   4  7/16     7/16
300  V  20   20    25  25  30  30  40  50  50  0  60       60
310  A  100  50    50  50  30  60  60  60  60  0  100      100
320  T  50   75    75  75  50  50  50  75  50  0  100      100
330
340 END
```

100 TEMPO 4=126

This TEMPO statement establishes a beginning tempo of 126 quarter notes per minute on both notelists.

110 AT BEAT 7 RITARD TO 4=72 IN 2

This statement causes a ritardando to begin on the seventh beat (the C4 in the first notelist and the A in the second notelist). The words AT BEAT are followed by a beat number (7), the words RITARD TO, and then the new tempo (4=72). Next, the word IN is followed by a beat number (2). This last expression causes the ritardando to occur over the length of two beats (the four eighth notes in the first notelist and the A and the G in the second notelist).

120 AT BEAT 9 TEMPO 4=126

This statement returns the composition to its original tempo on the ninth beat (the rest in both notelists). We could have used either the expression RITARD TO or the expression ACCEL TO in place of the word TEMPO in this statement and produced the same results. The abruptness of the change is due solely to the absence of the IN expression.

Linear or Logarithmic Tempo Changes

When specifying a gradual tempo change, you may also choose between linear and logarithmic change by entering the words LIN or LOG before the words ACCEL TO or RITARD TO. Logarithmic changes approach the new tempo more quickly at first. For example, we could have specified a logarithmic tempo change to begin at the seventh beat by entering this statement.

```
130 AT BEAT 7 LOG RITARD TO 4=72 IN 2
```

If you don't specify LIN or LOG (as we did not in TEMPO), a linear change will be automatically applied.

SYNCHRONIZATION STATEMENTS

There are three SCRIPT statements that allow you to synchronize a SCRIPT composition with specific time or frame cues. You can use these statements with compositions in SCRIPT music notation or in the computer music format. Thus, you can use the SCRIPT reverse compiler along with these statements to synchronize a sequence created on the Synclavier® II keyboard with film or video cues.

The synchronization statements can be used to cue a beat or a composition time with a different actual elapsed time cue, expressed in seconds or in frames Composition time is time as written in the composition that is used as a synchronization point.

The synchronization statements are grouped at the beginning of the composition after the overall tempo control statements. They divide the composition up into time segments. A time segment is the time between the composition time specified in one statement and the composition time specified in the previous statement. The first time segment is the time between the beginning of the composition and the composition time specified in the first synchronization statement.

You may group up to 100 synchronization statements together, thus creating 100 time segments, and you may intermix the functions as desired. You must enter the statements so that time values and beat numbers are listed in the order of occurrence.

DELAY

The DELAY statement causes the compiler to delay the start of the next segment in order to synchronize the composition time to the actual time. It inserts a rest before that point if no note is sounding. The DELAY statement may take either of two forms:

```
DELAY composition time TO actual elapsed time
or
DELAY BEAT beat number TO actual elapsed time
```

Composition time or actual elapsed time may be expressed in any of the time formats, in seconds or in frames, explained in the chapter on "SCRIPT Computer Music Format".

SYNC

The SYNC statement causes the SCRIPT compiler to adjust the tempo of the notes in the time segment so that the composition time or beat will be synchronized with the new actual elapsed time. The SYNC statement may take either of two forms:

```
SYNC composition time TO actual elapsed time
or
SYNC BEAT beat number TO actual elapsed time
```

SMOOTH

The SMOOTH statement causes a tempo change in the time segment, as does the SYNC statement. But the tempo change will occur gradually over the length of the segment. The SMOOTH statement may take either of two forms:

> SMOOTH composition time TO actual elapsed time
>
> or
>
> SMOOTH BEAT beat number TO actual elapsed time

Extreme tempo changes should be avoided in SMOOTH statements. Otherwise, oscillations in tempo will become audible.

Synchronizing a Composition in Music Notation

To show you how to synchronize a composition in music notation with time cues in frames, we have added two synchronization statements to TEMPO. This example is stored on your user diskette under file name SYNC1.

```
100 TEMPO 4=126
110 AT BEAT 7 RITARD TO 4=72 IN 2
120 AT BEAT 9 TEMPO 4=126
130
140 FPS 24
150 DELAY BEAT 1 TO :12
160 SYNC BEAT 7 TO :3:12
170
180 KEYSIG D
190 TIMBRE 75
200
210 NOTELIST USING 8-5
220
230  P  F   G   A   B   B   B   {C4 D    E}   C   R D
240  R  4   4   2   (4T)3        8  8    8    8   4 7/16
250  V  20  25  30  40  45  50  70  75   80   60  0 80
260  A  60  60  60  30  30  30  100 100  100  25  0 100
270
280 END
290
300 NOTELIST USING 8-5
310
320  P  A3   D    E   F   F   G   G#  A   G   R   [ F     A  ]
330  R  32 4-32   4   4.  8   4   4   4   4   4    7/16  7/16
340  V  20   20   25  25  30  30  40  50  50  0    60    60
350  A  100  50   50  50  30  60  60  60  60  0    100   100
360  T  50   75   75  75  50  50  50  75  50  0    100   100
370
380 END
```

Note that the composition begins with the overall tempo control statements. Since the following synchronization statements are based on beats, they must be preceded by a tempo statement so that the computer will know the rhythmic value of the beat.

140 FPS 24

> Since we are cueing with frames, we have included a frame rate statement.

150 DELAY BEAT 1 TO :12

> This statement delays the first beat to frame 12.

160 SYNC BEAT 7 TO :3:12

> This statement synchronizes the seventh beat (the C4 in the first notelist and the A in the second) with second 3, frame 12. The SCRIPT compiler calculates a new, slower, tempo for beats one through six.

Synchronizing a Composition in the Computer Music Format

Our second example takes the Bach Invention 13, which is in the computer music format, and synchronizes it with time cues in seconds. This example is stored on your user diskette under the file name SYNC2.

```
100 DELAY 4 TO 4.5
110 SYNC 8 TO 8.5
120 SYNC 12 TO 12
130 SYNC 16 TO 16
140 SMOOTH 20 TO 19
150
160 ARTICULATION MULTIPLIER 90
170
180 INSERT 'BACH13'
```

In this example, we have divided the composition into its eight measures; each measure lasts four seconds in original composition time. You might want to look at the listing of BACH13 in the appendix while you read the following explanation.

100 DELAY 4 TO 4.5

> This statement delays the first note of the second measure by one half second.

110 SYNC 8 TO 8.5

> This statement synchronizes the beginning of the third measure (8 seconds in composition time) with 8.5 seconds actual time. There is no tempo change in the second measure.

120 SYNC 12 TO 12

> This statement creates a faster tempo in the third measure (the notes from 8 to 12 seconds composition time) so that they will be completed by 12 seconds actual time.

130 SYNC 16 TO 16

> This statement causes the fourth measure to be played at the original tempo.

140 SMOOTH 20 TO 19

> This statement causes the fifth measure to be played with an accelerando so that 20 seconds composition time will be played at 19 seconds actual time.
>
> The sixth and seventh measures continue with the same tempo as the end of the fifth.

160 ARTICULATION MULTIPLIER 90

> Since this piece is legato, there are no spaces between the notes. In order to create a rest with our DELAY statement, we have shortened the durations of all notes slightly by adding an ARTICULATION MULTIPLIER statement.

180 INSERT 'BACH13'

> Note the use of the INSERT statement to insert an entire composition.

Synchronizing a Keyboard Performance

To synchronize a keyboard performance to film or video cues, follow this suggested method.

1. Run the Synclavier® II real-time system by typing PLAY after loading the SCRIPT system.
2. Choose a click rate which approximates the tempo you wish to use.
3. Create your composition on the Synclavier® II keyboard.
4. Use the reverse compiler to convert the sequence into the computer music format at the proper frame rate.
5. Read your cues from the SMPTE reader or film synchronizer.
6. Choose notes in the composition which you wish to align with these cues.
7. Use the synchronization statements to align these notes with the cues. The SCRIPT compiler will calculate the exact tempos, accurate to better than one frame in an hour of elapsed time, to perform all notes between synchronization statements. Notice that a constant tempo is not necessary since multiple synchronization statements are possible.
8. Type PLAY again.

SMPTE "color time"

SCRIPT can easily maintain long-term synchrony with SMPTE time and control code when the color frame rate is 29.94 fps. You may use a SCRIPT frame rate of 30 fps in all cases.

If the SMPTE color time is encoded in the drop frame mode, no correction is required to assure long-term synchrony within 2 frames.

If the SMPTE color time is encoded in the uncompensated mode, an exact correspondence between SMPTE frame numbers and SCRIPT frame numbers may be established by the following SYNC statement:

```
SYNC :1:00:00:0 to :1:00:00:108
```

This statement will precisely compensate for the SMPTE color time error of +108 frames per hour of elapsed time (+3.6 seconds per hour of elapsed time).

This chapter contains information about user files. In it, you will learn how to use Synclavier® II timbres and sequences with the SCRIPT system, how to format and copy diskettes, how to increase the number of files that can be stored on a diskette, and how to print permanent copies of SCRIPT compositions.

USER AND SYSTEM FILES

Different sized diskettes have different storage capacities. As of Release H, the capacities are as follows:

	User Files	Words (approx.)
Single density 5¼ inch	29	40,000
Double density 5¼ inch	30	200,000
Single density 8 inch	31	300,000

The term words here refers to computer storage words, each containing two characters of text. The CATALOG and LENGTH monitor commands also refer to lengths measured in words. These numbers will change in future releases.

In addition to the user files, SCRIPT user diskettes also contain special "system" files. Because the names of these files begin with periods (.), their names are not listed by the CATALOG command. These system files are used to store Synclavier® II timbre banks and sequences. Thus, timbre and sequence information may exist in two forms on the SCRIPT user diskettes: (1) in system files, from which they may be accessed by the STORE and RECALL buttons during Synclavier® II operation, and (2) in SCRIPT text files.

SCRIPT TIMBRE DEFINITIONS AND SYNCLAVIER® II TIMBRE BANKS

Using a Synclavier® II Timbre in a SCRIPT Composition

A timbre in a timbre bank on the SCRIPT user diskette may be automatically included in a SCRIPT composition through use of the NOTELIST USING statement.

Also, during Synclavier® II operation, any timbre in the timbre banks may be substituted for the timbre on any of the tracks of the memory recorder through use of the SMT button; see the instructions in the *Synclavier® II Instruction Manual.* (By inserting different Synclavier® II timbre diskettes in the right-hand drive, you may use any Synclavier® II timbre in your composition.)

Storing a SCRIPT Timbre Definition in a Synclavier® II Timbre Bank

To store a timbre created by a SCRIPT timbre definition in a Synclavier® II timbre bank, first PLAY the composition containing a notelist which uses the timbre. Then, during Synclavier® II operation, follow these instructions:

1. Press the TIMBRE BANK button for the bank in which you want to store the timbre.
2. Press the SKT button.
3. Press the TRACK button corresponding to the track which has the timbre you want to store. (The timbres will be located on the tracks in the order in which they were used in the notelists of the SCRIPT composition.) The timbre will now be active on the Synclavier® II keyboard. To be sure you have the right timbre, play a few notes on the keyboard.

4. While holding down the ENTRY WRITE button, press the TIMBRE ENTRY button corresponding to the entry number in which you wish to store the timbre. This will store the timbre on the diskette.

SCRIPT COMPOSITIONS AND SYNCLAVIER® II SEQUENCES

While the information in SCRIPT text files is organized for convenient display and editing from the terminal, the information in a Synclavier® II sequence file is organized very compactly for rapid access during real-time Synclavier® II operation.

During Synclavier® II operation, sequences are stored and recalled from Synclavier® II sequence diskettes by use of the RECORDER STORE/RECALL buttons. In maxidiskette systems, two Synclavier® II sequences may also be stored on the SCRIPT user diskette.

For your convenience, SCRIPT includes three monitor commands which allow you to CONVERT a text composition into a sequence, to STORE this sequence in a Synclavier® II sequence file, and to RECALL the sequence from the sequence file.

Once a sequence is stored on a Synclavier® II sequence diskette, it may be played on any Synclavier® II system with enough computer memory to hold the number of notes in the sequence, even if the system is not equipped with a computer terminal or SCRIPT. A Release E (or later release) Synclavier® II operating system diskette should be used to play sequences compiled by SCRIPT.

The CONVERT Command

When you type the monitor command CONVERT, the SCRIPT compiler converts the current file (which should contain a composition in text form) into a Synclavier® II sequence. The converted sequence becomes the new current file, and the text in the previous current file is lost. The name of the current file is automatically changed by adding a period (.) to the end of the current file name. This precaution insures that you will not accidentally REPLACE your text file with the converted file.

A converted sequence may be saved on a user diskette under any valid file name. Then when you type OLD <filename> and PLAY, the composition will play without the delay which would otherwise be required for conversion. If you wish to be able to recall the converted sequence during Synclavier® II real-time operation with a STORE/RECALL button, you must first store it in a Synclavier® II sequence file with the STORE command.

The STORE Command

The STORE command stores the current file (which should contain a converted sequence) in a Synclavier® II sequence file. Use the word STORE followed by a sequence file number (the number of one of the buttons under RECORDER STORE/RECALL). To store a converted sequence in sequence file number 2, for example, first insert a Synclavier® II sequence diskette in the right-hand drive; then type

```
STORE 2
```

Any number from 1 to 6 may be used, but not all sequence diskettes can store 6 sequences. An error message will be printed on the terminal if a Synclavier® II sequence file with that number does not exist on the diskette in the right-hand drive, or if there are too many notes in your converted sequence to fit into the sequence file.

The RECALL Command

The RECALL command recalls a sequence from a Synclavier® II sequence file and places it in the current file. The number of the sequence file will become the name of the current file and the previous contents of the current file will be lost. Use the word RECALL followed by a sequence file number. To recall the converted sequence from sequence file number 2, for example, from a Synclavier® II sequence diskette, first insert the diskette into the right-hand drive, then type

```
RECALL 2
```

The current file will then contain the sequence and will be named "2". A SCRIPT user diskette can now be placed in the right-hand drive and the sequence can be saved with the name "2", or any other valid file name.

DUPLICATING SCRIPT FILES ONTO OTHER USER DISKETTES

You may easily copy a file on your SCRIPT user diskette onto another user diskette; the OLD and SAVE commands can be used to transfer one file at a time. For example, to copy a file named MELODY, follow this procedure:

1. Type OLD MELODY and press RETURN.
2. Remove the user diskette from the right-hand drive and replace it with the user diskette on which you wish to make the copy.
3. Type SAVE

USING THE FORMCOPY UTILITY PROGRAM

You can also duplicate all files on a diskette (including the "hidden" system files) with the FORMCOPY Utility Program. Thus, you can create new copies of any SCRIPT or Synclavier® II system or user diskette.

If you have purchased blank, unformatted diskettes, they must be **formatted** before you can write any information on them. The FORMCOPY program allows you to format a new diskette and then copy the information from another diskette onto it.

The FORMCOPY program is stored on the special XPL/4 Utility Program Diskette. (If you are using single density minidiskettes FORMCOPY is on the Vol. I Utility Programs Diskette.)

Step-by-step instructions are printed on the screen on the terminal screen when FORMCOPY is run. Further instructions on how to use FORMCOPY may be found in the *Scientific XPL/4 Documentation Update* (Section 9 in this binder.)

This is the basic procedure:

1. Insert your Utility Programs Diskette in the right-hand drive.
2. Type

   ```
   OLD FORMCOPY
   RUN
   ```

   The monitor command RUN is used to run a computer program located in the current file.
3. Remove the Utility Programs Diskette from the right-hand drive and the SCRIPT system diskette from the left-hand drive.
4. Insert the new blank diskette in the **left-hand** drive and the diskette you wish to copy in the **right-hand** drive.
5. Type Y for FORMAT and Y for COPY.

6. Then type GO (after making sure the right diskettes are in the right drives).
7. Then press RETURN in response to the safety prompt.

   The formatting and copying will be performed.

   When done, you will be asked if you wish to format and copy more diskettes. If not, type STOP and replace your SCRIPT system diskette.

CREATING ROOM FOR MORE FILES ON A USER DISKETTE

Running Out of Room on a User Diskette

After saving many SCRIPT compositions on your user diskette, you may get an error message printed on your terminal when you try to replace or save a file. You then have three choices: One, you can unsave unnecessary files and then save your current file. Two, you can insert a new user diskette with sufficient space on which you can save your current file. Or three, you can use the SHUFFLE program.

Using the SHUFFLE Utility Program

In many cases, when you get the message that you are out of room, sufficient unused space exists on the diskette, but it is fragmented into many short segments located between user files. To remove these "holes", you may run the SHUFFLE program on the Utility Programs Diskette (Vol. II for single density minidiskettes). Follow the instructions on the screen and see the *Scientific XPL/4 Documentation Update* for more details.

PERMANENT LISTINGS OF SCRIPT COMPOSITIONS

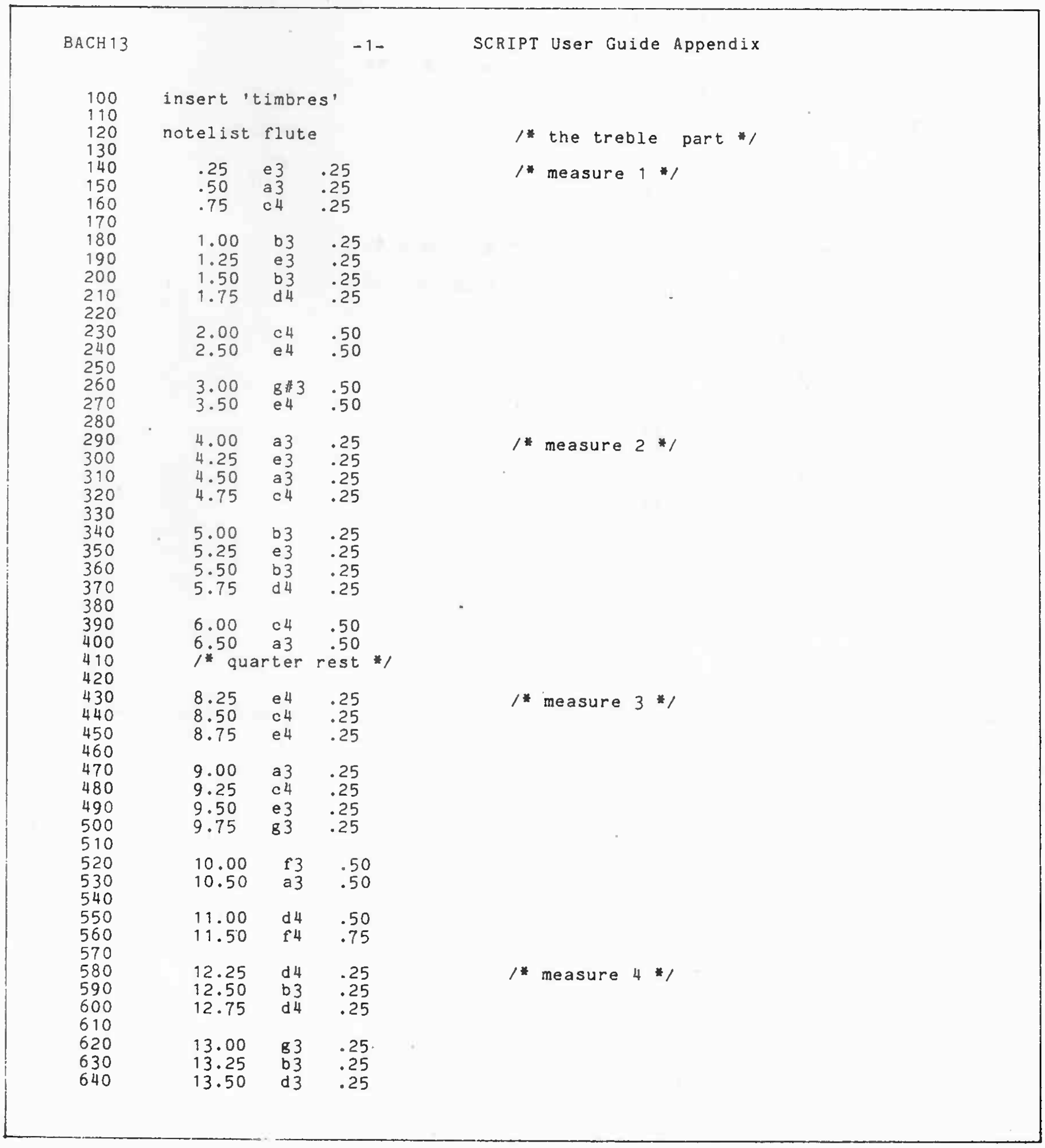
If you have a hard-copy terminal, or if you have the Printer/Modem Port Option and a printer, you may make permanent listings of your SCRIPT compositions with the LISTING program on the Utility Programs diskette (VOL. II for single density minidiskettes). The LISTING program prints the listing in a more readable format than that produced by the PRINT monitor command. Follow the instructions on the screen and see the *Scientific XPL/4 Documentation Update* for more details.

APPENDIX

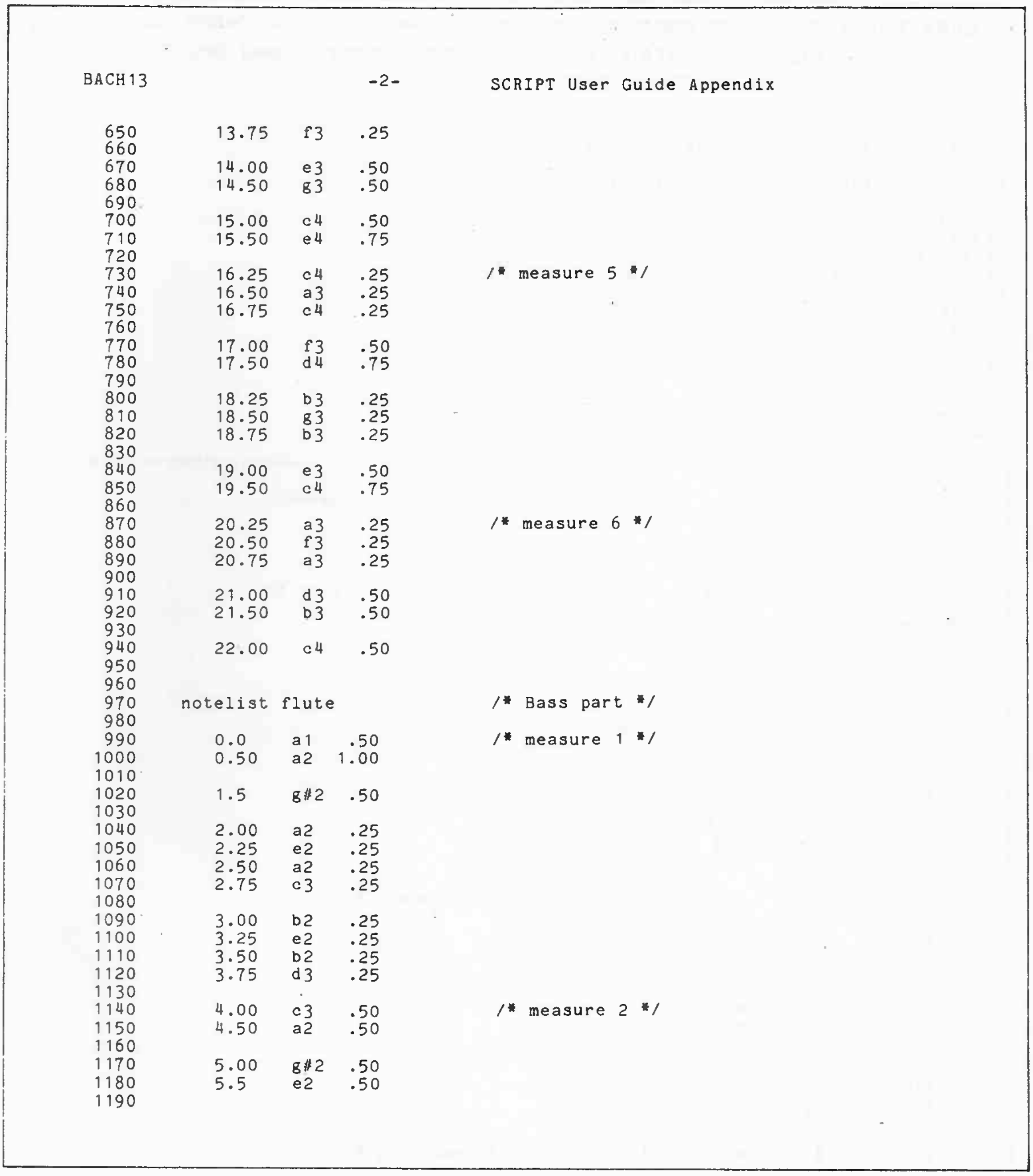
Please Note: The page numbers in the following index refer to the page numbers listed in the printout reproduced below, not to the pages of this manual.

INDEX SCRIPT User Guide Appendix

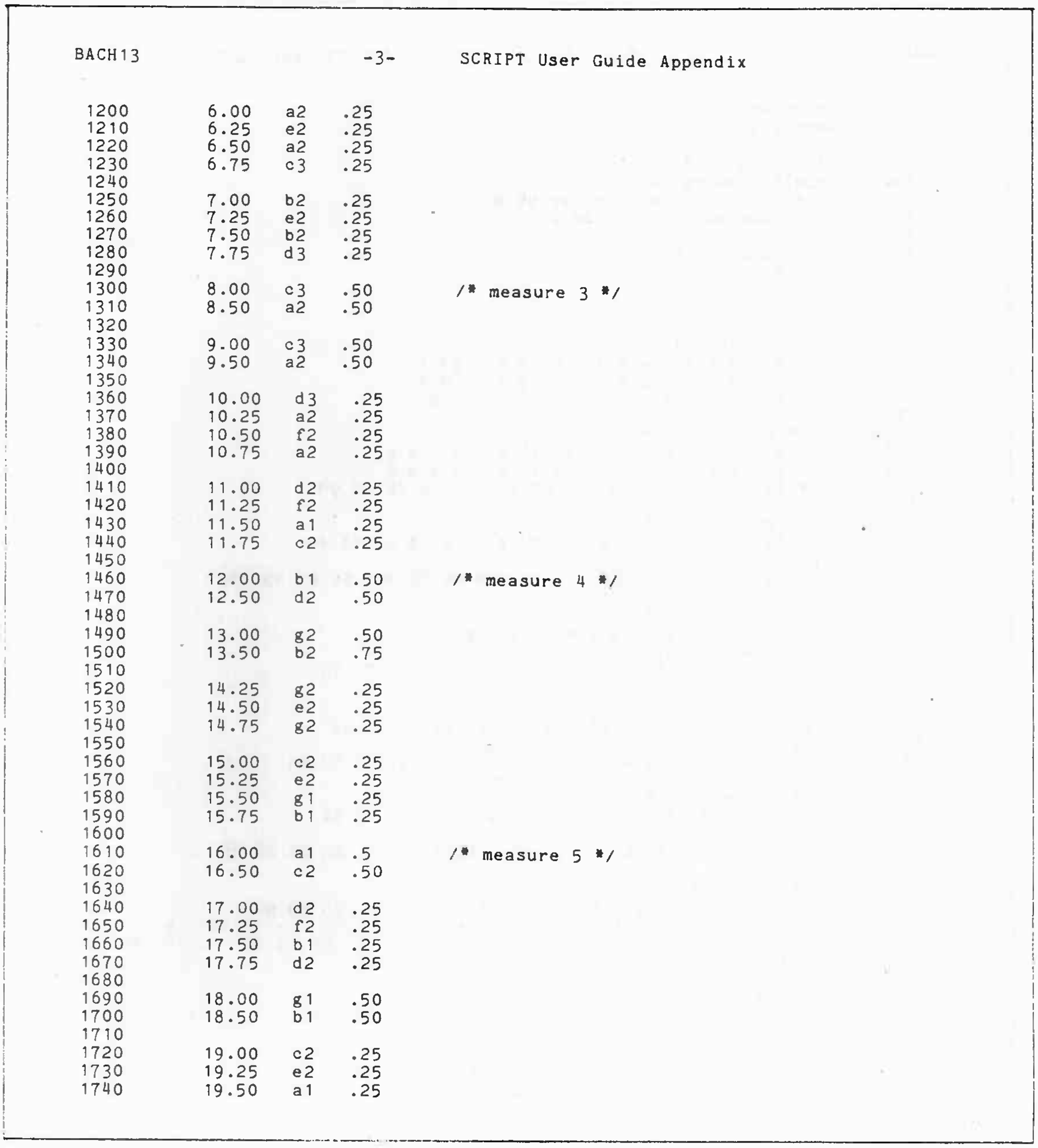
Filename	Page	Used Sectors: 260
Bach13	1	1669
Brand3	5	1538
Dynamics	7	305
Errors	8	153
Key	9	110
Phrase	10	153
Polly	11	476
Rhythm	12	143
Sync1	13	365
Sync2	14	64
Tempo	15	333
Timbres	16	149
Last Page	16	



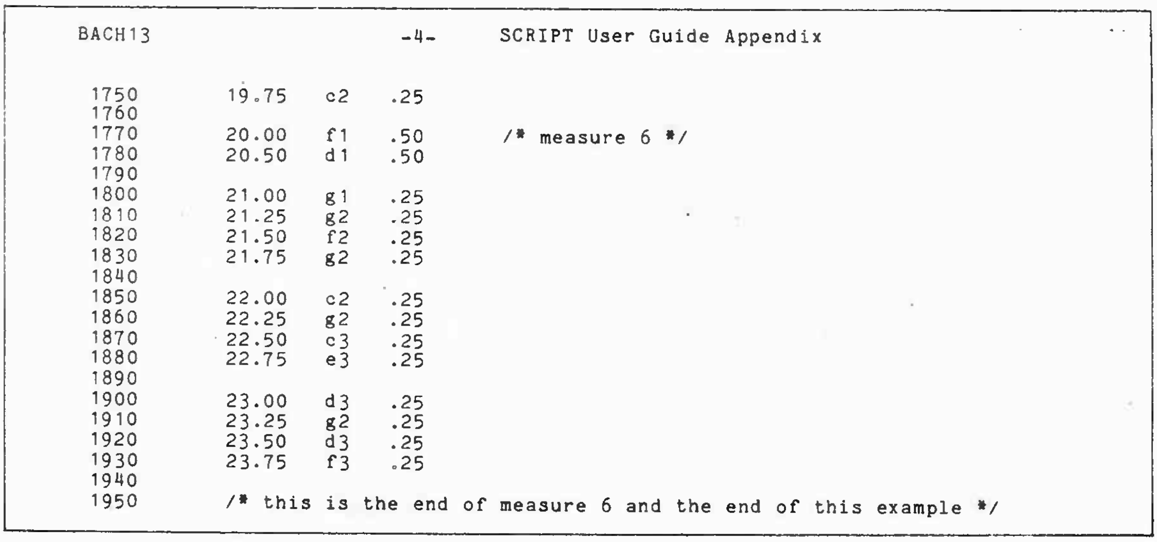
```
BACH13                          -1-          SCRIPT User Guide Appendix


     100    insert 'timbres'
     110
     120    notelist flute                        /* the treble  part */
     130
     140       .25    e3   .25                    /* measure 1 */
     150       .50    a3   .25
     160       .75    c4   .25
     170
     180       1.00    b3   .25
     190       1.25    e3   .25
     200       1.50    b3   .25
     210       1.75    d4   .25
     220
     230       2.00    c4   .50
     240       2.50    e4   .50
     250
     260       3.00    g#3  .50
     270       3.50    e4   .50
     280
     290       4.00    a3   .25                   /* measure 2 */
     300       4.25    e3   .25
     310       4.50    a3   .25
     320       4.75    c4   .25
     330
     340       5.00    b3   .25
     350       5.25    e3   .25
     360       5.50    b3   .25
     370       5.75    d4   .25
     380
     390       6.00    c4   .50
     400       6.50    a3   .50
     410       /* quarter rest */
     420
     430       8.25    e4   .25                   /* measure 3 */
     440       8.50    c4   .25
     450       8.75    e4   .25
     460
     470       9.00    a3   .25
     480       9.25    c4   .25
     490       9.50    e3   .25
     500       9.75    g3   .25
     510
     520       10.00    f3   .50
     530       10.50    a3   .50
     540
     550       11.00    d4   .50
     560       11.50    f4   .75
     570
     580       12.25    d4   .25                  /* measure 4 */
     590       12.50    b3   .25
     600       12.75    d4   .25
     610
     620       13.00    g3   .25
     630       13.25    b3   .25
     640       13.50    d3   .25
```



```
 650      13.75   f3   .25          /* measure 5 */
 660
 670      14.00   e3   .50
 680      14.50   g3   .50
 690
 700      15.00   c4   .50
 710      15.50   e4   .75
 720
 730      16.25   c4   .25          /* measure 5 */
 740      16.50   a3   .25
 750      16.75   c4   .25
 760
 770      17.00   f3   .50
 780      17.50   d4   .75
 790
 800      18.25   b3   .25
 810      18.50   g3   .25
 820      18.75   b3   .25
 830
 840      19.00   e3   .50
 850      19.50   c4   .75
 860
 870      20.25   a3   .25          /* measure 6 */
 880      20.50   f3   .25
 890      20.75   a3   .25
 900
 910      21.00   d3   .50
 920      21.50   b3   .50
 930
 940      22.00   c4   .50
 950
 960
 970    notelist flute              /* Bass part */
 980
 990      0.0     a1   .50          /* measure 1 */
1000      0.50    a2  1.00
1010
1020      1.5     g#2  .50
1030
1040      2.00    a2   .25
1050      2.25    e2   .25
1060      2.50    a2   .25
1070      2.75    c3   .25
1080
1090      3.00    b2   .25
1100      3.25    e2   .25
1110      3.50    b2   .25
1120      3.75    d3   .25
1130
1140      4.00    c3   .50          /* measure 2 */
1150      4.50    a2   .50
1160
1170      5.00    g#2  .50
1180      5.5     e2   .50
1190
```

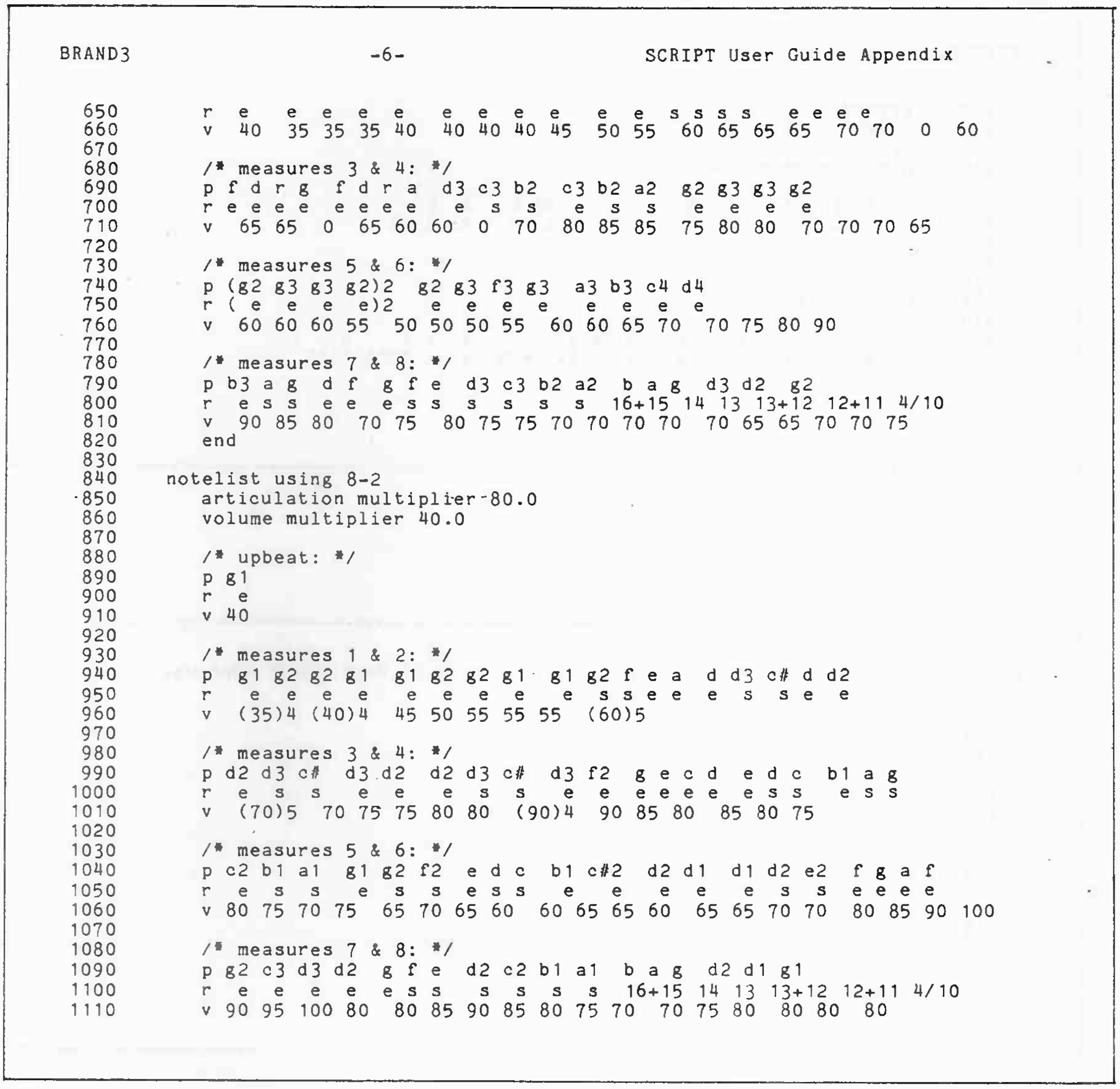


```
1200       6.00   a2   .25
1210       6.25   e2   .25
1220       6.50   a2   .25
1230       6.75   c3   .25
1240
1250       7.00   b2   .25
1260       7.25   e2   .25
1270       7.50   b2   .25
1280       7.75   d3   .25
1290
1300       8.00   c3   .50          /* measure 3 */
1310       8.50   a2   .50
1320
1330       9.00   c3   .50
1340       9.50   a2   .50
1350
1360       10.00   d3   .25
1370       10.25   a2   .25
1380       10.50   f2   .25
1390       10.75   a2   .25
1400
1410       11.00   d2   .25
1420       11.25   f2   .25
1430       11.50   a1   .25
1440       11.75   c2   .25
1450
1460       12.00   b1   .50        /* measure 4 */
1470       12.50   d2   .50
1480
1490       13.00   g2   .50
1500       13.50   b2   .75
1510
1520       14.25   g2   .25
1530       14.50   e2   .25
1540       14.75   g2   .25
1550
1560       15.00   c2   .25
1570       15.25   e2   .25
1580       15.50   g1   .25
1590       15.75   b1   .25
1600
1610       16.00   a1   .5         /* measure 5 */
1620       16.50   c2   .50
1630
1640       17.00   d2   .25
1650       17.25   f2   .25
1660       17.50   b1   .25
1670       17.75   d2   .25
1680
1690       18.00   g1   .50
1700       18.50   b1   .50
1710
1720       19.00   c2   .25
1730       19.25   e2   .25
1740       19.50   a1   .25
```



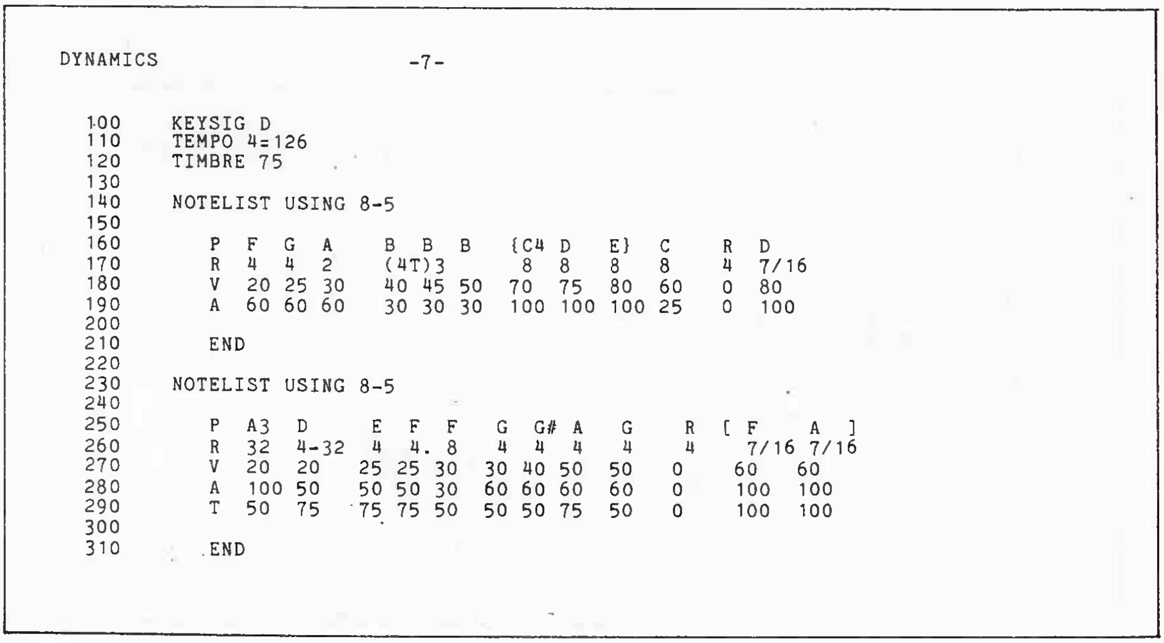
```
1750       19.75   c2   .25
1760
1770       20.00   f1   .50        /* measure 6 */
1780       20.50   d1   .50
1790
1800       21.00   g1   .25
1810       21.25   g2   .25
1820       21.50   f2   .25
1830       21.75   g2   .25
1840
1850       22.00   c2   .25
1860       22.25   g2   .25
1870       22.50   c3   .25
1880       22.75   e3   .25
1890
1900       23.00   d3   .25
1910       23.25   g2   .25
1920       23.50   d3   .25
1930       23.75   f3   .25
1940
1950       /* this is the end of measure 6 and the end of this example */
```

```
tempo 4=85
keysig g

/* violin I: */
notelist using 8-1
   articulation multiplier 90.0
   volume multiplier 100.0

   /* upbeat: */
   p g f
   r s s
   v 35 40

   /* measure 1: */
   p g   d c d   g f g   b2 a b   g3 f
   r e   s s e   s s e   s s e    s s
   v 45  35 40 45  (45 50 55)2  55 60

   /* measure 2: */
   p g  g2 a b c#3   {d c# d} e   d f d g
   r e   s s e   e    s  s s  s   s s s s
   v 70  60 65 70  75  70 65 65 70  70 75 80 90

   /* measure 3: */
   p {d c# d} e  d a d b  {d c# d} e  d c4 d3 d4
   r (s)16
   v 75 70 70 75  80 75 85 90  75 70 70 75  85 80 90 100

   /* measure 4: */
   p b3 a g  a g f  g f e  d g d g
   r (e s s)3 (s)4
   v 100 95 85  90 85 75  80 70 65  65 70 70 75

   /* measures 5: */
   p e d c  b2 g3 b2 g3  c3 b2 a2  g2 g3 a2 g3
   r(e(s)6)2
   v  75 65 60  65 65 65 65  65 60 60  50 50 55 55

   /* measure 6: */
   p b2 g3 c# g3  d f d g  d a d b  d3 c4 d3 d4
   r (s)16
   v  50 55 55 57  60 65 65 67  70 75 75 80  80 85 85 90

   /* measures 7 & 8: */
   p b3 a g  d f  g f e  d3 c3 b2 a2  b a g  d3 d3 g2
   r  e s s  e e  e s s  s  s  s  s 16+15 14 13 13+12 12+11 4/10
   v  90 85 80  70 75  80 75 70  70 70 65 65  65 70 70  70 70  70
   end

notelist using 8-1
   articulation multiplier 90.0
   volume multiplier 60.0

   /* upbeat and measures 1 & 2: */
   p g2  b2 d3 b2 g2  b2 d3 b2 g2  b2 d3  g b a g   f d r b
```



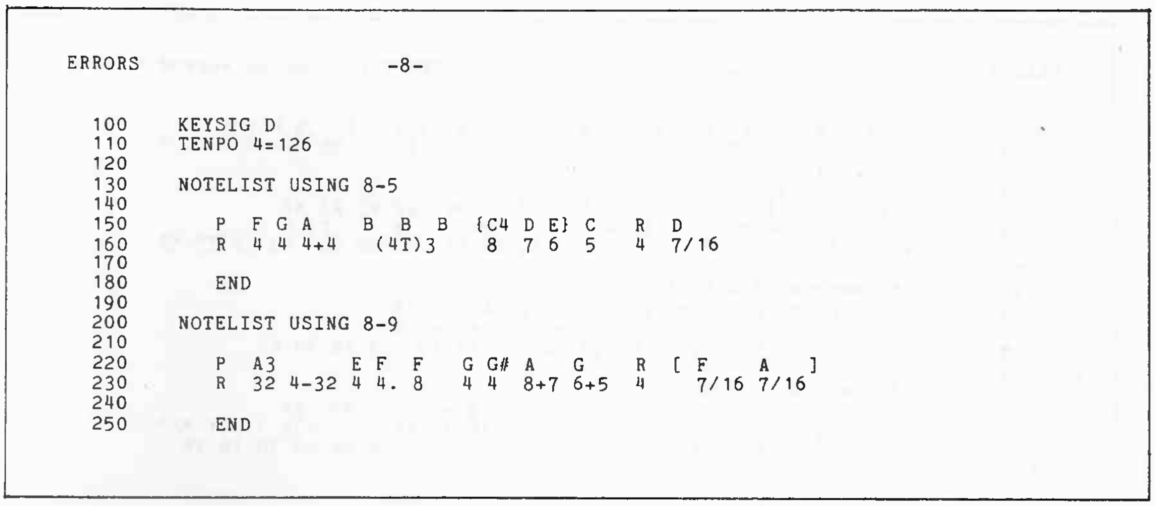
BRAND3 -6- SCRIPT User Guide Appendix

```
  650      r  e    e   e   e   e    e   e   e   e    e   e  s s s s   e e e e
  660      v  40   35 35  35 40   40 40  40 45   50 55   60 65 65 65   70 70   0   60
  670
  680      /* measures 3 & 4: */
  690      p f d r g  f d r a  d3 c3 b2  c3 b2 a2  g2 g3 g3 g2
  700      r e e e e  e e e e   e  s  s   e  s  s   e  e  e  e
  710      v  65 65  0  65 60 60  0  70  80 85 85  75 80 80  70 70 70 65
  720
  730      /* measures 5 & 6: */
  740      p (g2 g3 g3 g2)2  g2 g3 f3 g3  a3 b3 c4 d4
  750      r ( e  e  e  e)2   e  e  e  e   e  e  e  e
  760      v  60 60 60 55  50 50 50 55  60 60 65 70  70 75 80 90
  770
  780      /* measures 7 & 8: */
  790      p b3 a g  d f  g f e  d3 c3 b2 a2  b a g  d3 d2  g2
  800      r  e s s  e e  e s s  s  s  s  s  16+15 14 13 13+12 12+11 4/10
  810      v  90 85 80  70 75  80 75 75 70 70 70 70  70 65 65 70 70 75
  820      end
  830
  840   notelist using 8-2
 ·850      articulation multiplier-80.0
  860      volume multiplier 40.0
  870
  880      /* upbeat: */
  890      p g1
  900      r  e
  910      v 40
  920
  930      /* measures 1 & 2: */
  940      p  g1 g2 g2 g1  g1 g2 g2 g1  g1 g2 f e a  d d3 c# d d2
  950      r   e  e  e  e   e  e  e  e   e  s s e e  e  s  s e  e
  960      v  (35)4 (40)4  45 50 55 55 55  (60)5
  970
  980      /* measures 3 & 4: */
  990      p d2 d3 c#  d3 d2  d2 d3 c#  d3 f2  g e c d  e d c  b1 a g
 1000      r  e  s  s   e  e   e  s  s   e  e  e e e e  e s s   e s s
 1010      v  (70)5  70 75 75 80 80  (90)4  90 85 80  85 80 75
 1020
 1030      /* measures 5 & 6: */
 1040      p c2 b1 a1  g1 g2 f2  e d c  b1 c#2  d2 d1  d1 d2 e2  f g a f
 1050      r  e  s  s   e  s  s  e s s   e   e   e  e   e  s  s  e e e e
 1060      v 80 75 70 75  65 70 65 60  60 65 65 60  65 65 70 70  80 85 90 100
 1070
 1080      /* measures 7 & 8: */
 1090      p g2 c3 d3 d2  g f e  d2 c2 b1 a1  b a g  d2 d1 g1
 1100      r  e  e  e  e  e s s   s  s  s  s  16+15 14 13 13+12 12+11 4/10
 1110      v 90 95 100 80  80 85 90 85 80 75 70  70 75 80  80 80  80
```

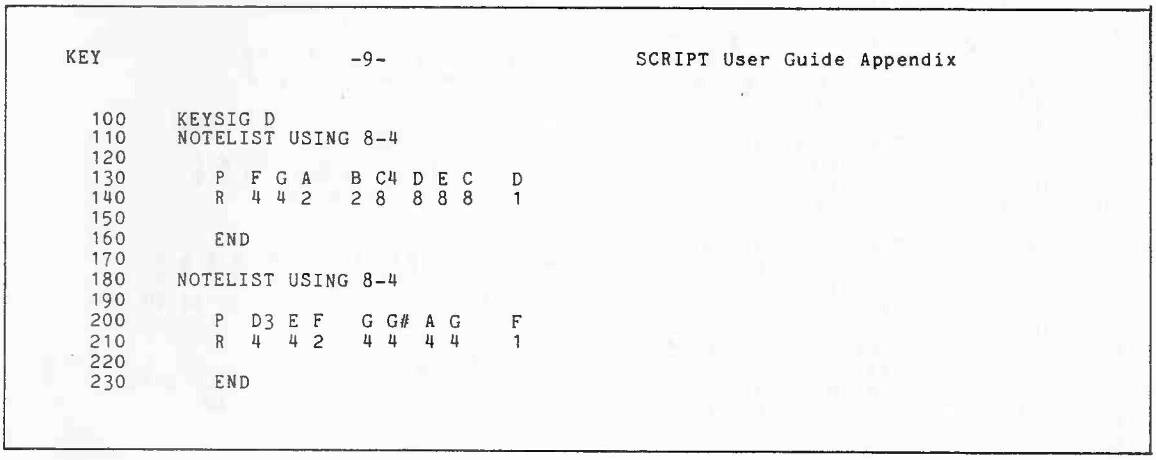


DYNAMICS -7-

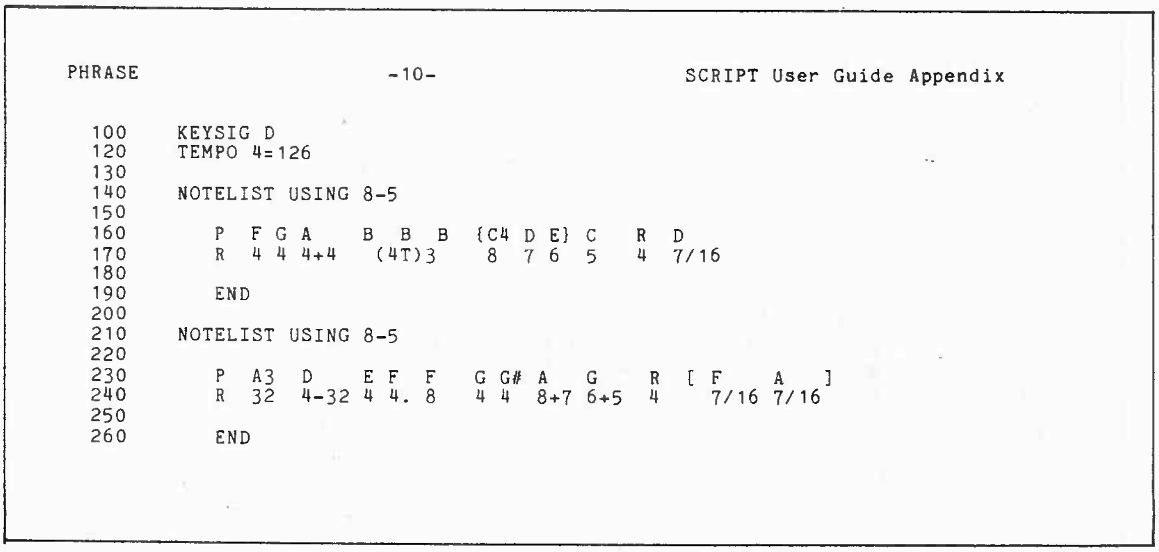
```
  100     KEYSIG D
  110     TEMPO 4=126
  120     TIMBRE 75
  130
  140     NOTELIST USING 8-5
  150
  160        P  F  G  A    B  B  B   {C4  D   E}  C    R  D
  170        R  4  4  2    (4T)3      8   8   8   8    4  7/16
  180        V  20 25 30   40 45 50  70  75  80  60    0  80
  190        A  60 60 60   30 30 30  100 100 100 25    0  100
  200
  210        END
  220
  230     NOTELIST USING 8-5
  240
  250        P  A3  D      E  F  F   G  G# A   G     R  [ F     A  ]
  260        R  32  4-32   4  4. 8   4  4  4   4     4    7/16 7/16
  270        V  20  20    25 25 30  30 40 50  50    0    60    60
  280        A  100 50    50 50 30  60 60 60  60    0    100   100
  290        T  50  75    75 75 50  50 50 75  50    0    100   100
  300
  310        END
```



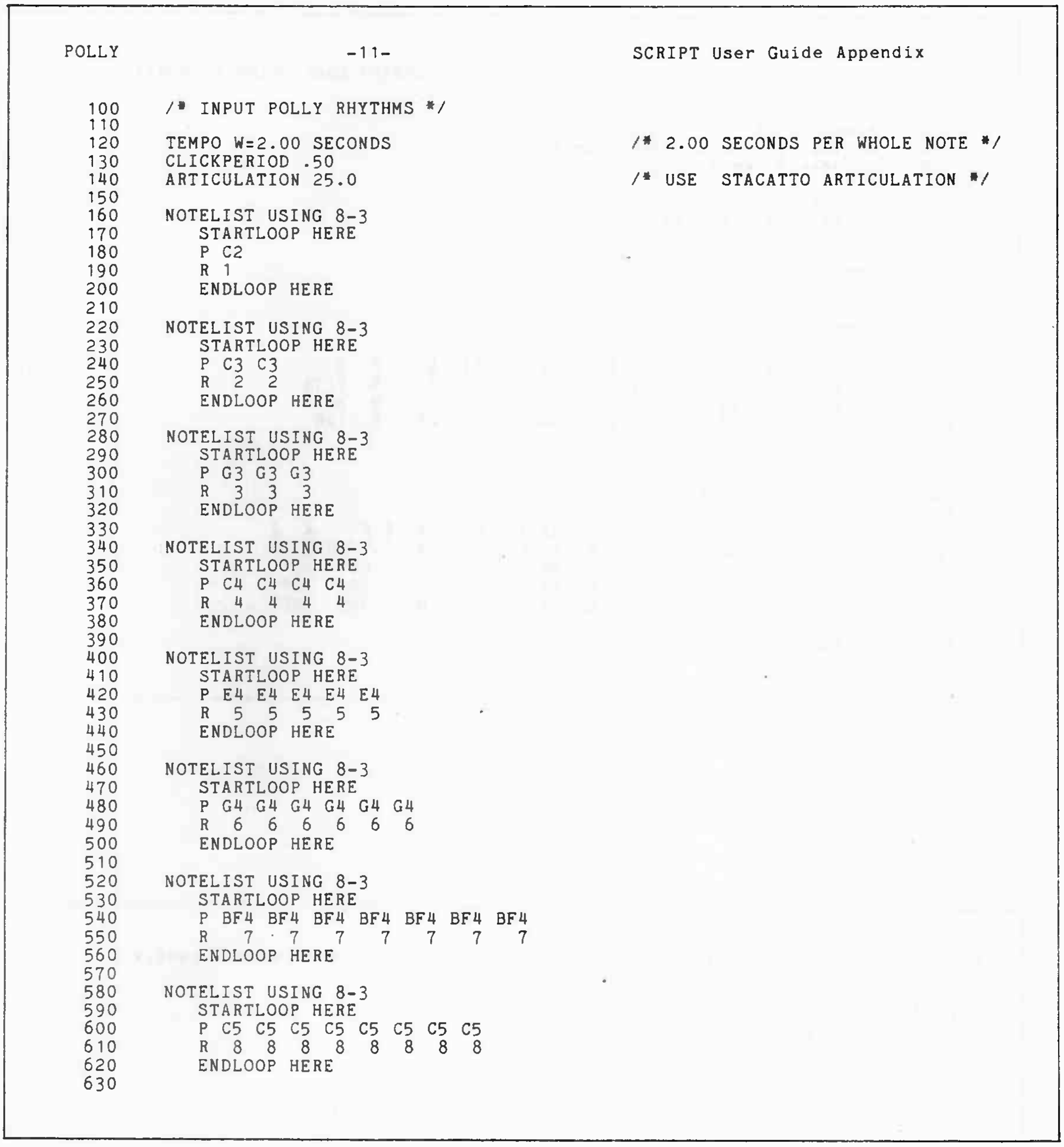
```
ERRORS                    -8-

  100    KEYSIG D
  110    TENPO 4=126
  120
  130    NOTELIST USING 8-5
  140
  150       P  F G A    B  B  B  {C4 D E} C   R  D
  160       R  4 4 4+4   (4T)3    8  7 6  5   4  7/16
  170
  180       END
  190
  200    NOTELIST USING 8-9
  210
  220       P  A3      E F  F   G G# A   G    R  [ F    A   ]
  230       R  32 4-32 4 4. 8   4 4  8+7 6+5  4    7/16 7/16
  240
  250       END
```



```
KEY                    -9-                    SCRIPT User Guide Appendix

  100    KEYSIG D
  110    NOTELIST USING 8-4
  120
  130       P  F G A   B C4 D E C   D
  140       R  4 4 2   2 8  8 8 8   1
  150
  160       END
  170
  180    NOTELIST USING 8-4
  190
  200       P  D3 E F   G G# A G    F
  210       R  4  4 2   4 4  4 4    1
  220
  230       END
```

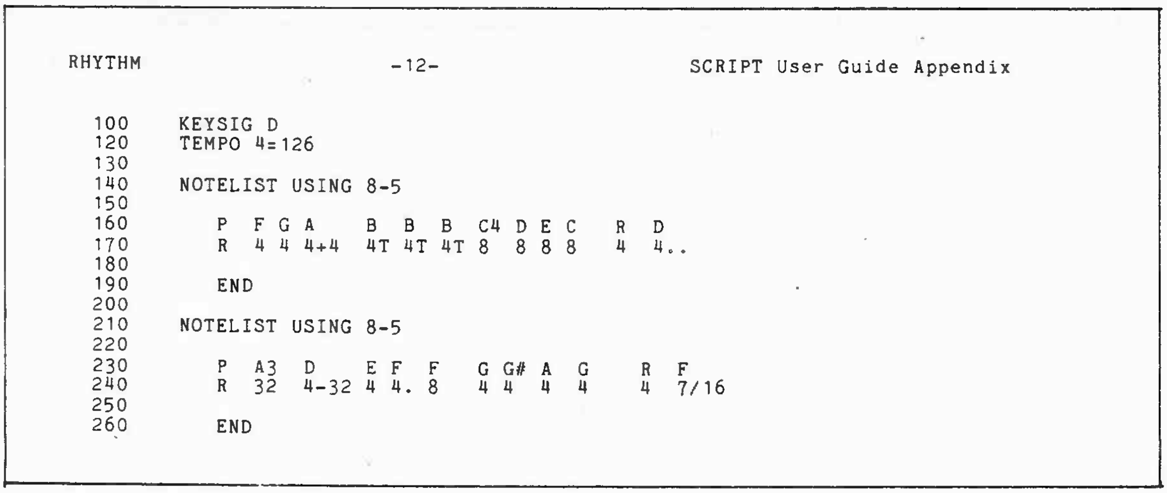


```
PHRASE                    -10-                    SCRIPT User Guide Appendix

  100    KEYSIG D
  120    TEMPO 4=126
  130
  140    NOTELIST USING 8-5
  150
  160       P  F G A    B  B  B  {C4 D E} C   R  D
  170       R  4 4 4+4   (4T)3    8  7 6  5   4  7/16
  180
  190       END
  200
  210    NOTELIST USING 8-5
  220
  230       P  A3  D    E F  F   G G# A   G    R  [ F    A   ]
  240       R  32  4-32 4 4. 8   4 4  8+7 6+5  4    7/16 7/16
  250
  260       END
```



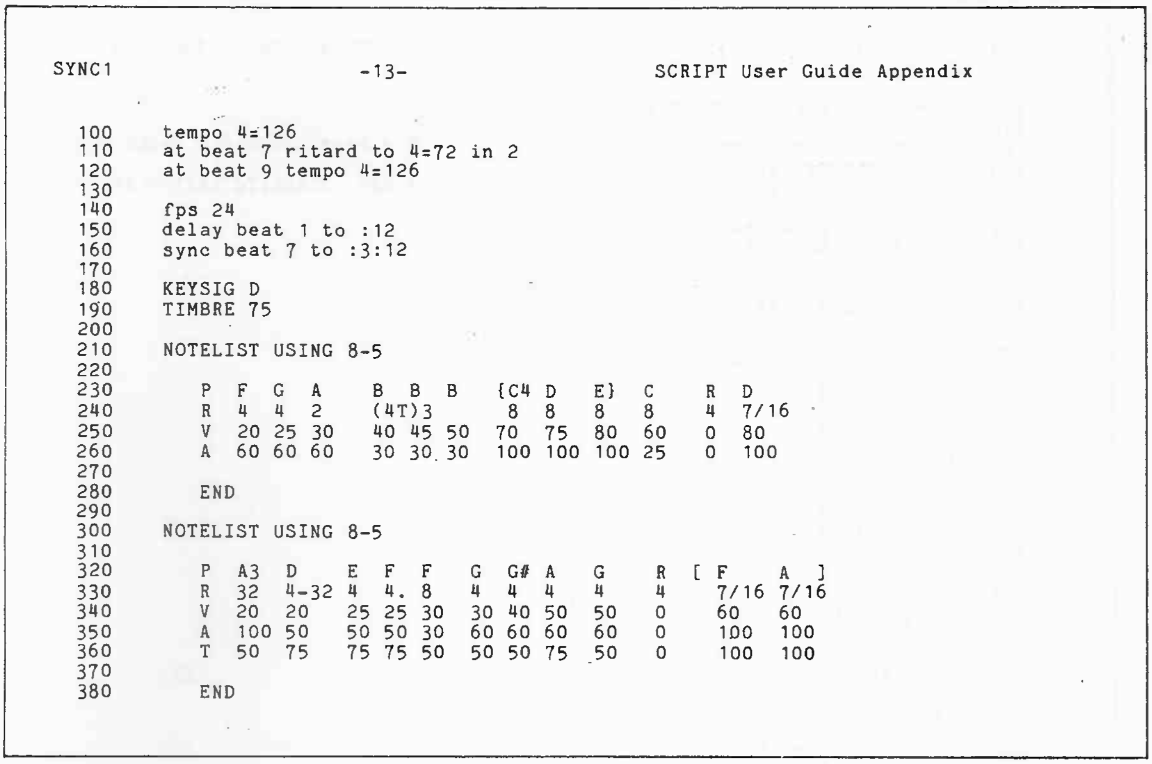
POLLY   -11-   SCRIPT User Guide Appendix

```
/* INPUT POLLY RHYTHMS */

TEMPO W=2.00 SECONDS                         /* 2.00 SECONDS PER WHOLE NOTE */
CLICKPERIOD .50
ARTICULATION 25.0                            /* USE  STACATTO ARTICULATION */

NOTELIST USING 8-3
   STARTLOOP HERE
   P C2
   R 1
   ENDLOOP HERE

NOTELIST USING 8-3
   STARTLOOP HERE
   P C3 C3
   R  2  2
   ENDLOOP HERE

NOTELIST USING 8-3
   STARTLOOP HERE
   P G3 G3 G3
   R  3  3  3
   ENDLOOP HERE

NOTELIST USING 8-3
   STARTLOOP HERE
   P C4 C4 C4 C4
   R  4  4  4  4
   ENDLOOP HERE

NOTELIST USING 8-3
   STARTLOOP HERE
   P E4 E4 E4 E4 E4
   R  5  5  5  5  5
   ENDLOOP HERE

NOTELIST USING 8-3
   STARTLOOP HERE
   P G4 G4 G4 G4 G4 G4
   R  6  6  6  6  6  6
   ENDLOOP HERE

NOTELIST USING 8-3
   STARTLOOP HERE
   P BF4 BF4 BF4 BF4 BF4 BF4 BF4
   R   7   7   7   7   7   7   7
   ENDLOOP HERE

NOTELIST USING 8-3
   STARTLOOP HERE
   P C5 C5 C5 C5 C5 C5 C5 C5
   R  8  8  8  8  8  8  8  8
   ENDLOOP HERE
```



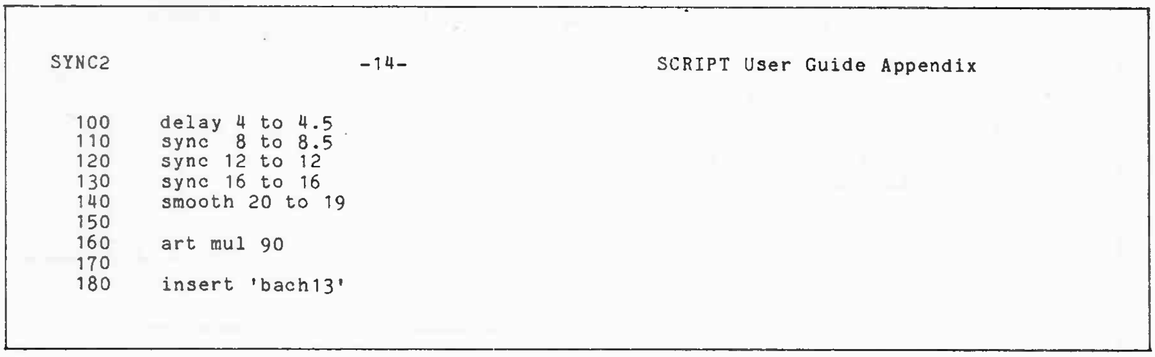
RHYTHM   -12-   SCRIPT User Guide Appendix

```
KEYSIG D
TEMPO 4=126

NOTELIST USING 8-5

   P  F G A    B  B  B  C4 D E C    R  D
   R  4 4 4+4  4T 4T 4T 8  8 8 8    4  4..

   END

NOTELIST USING 8-5

   P  A3  D    E F  F    G G# A  G    R  F
   R  32  4-32 4 4. 8    4 4  4  4    4  7/16

   END
```



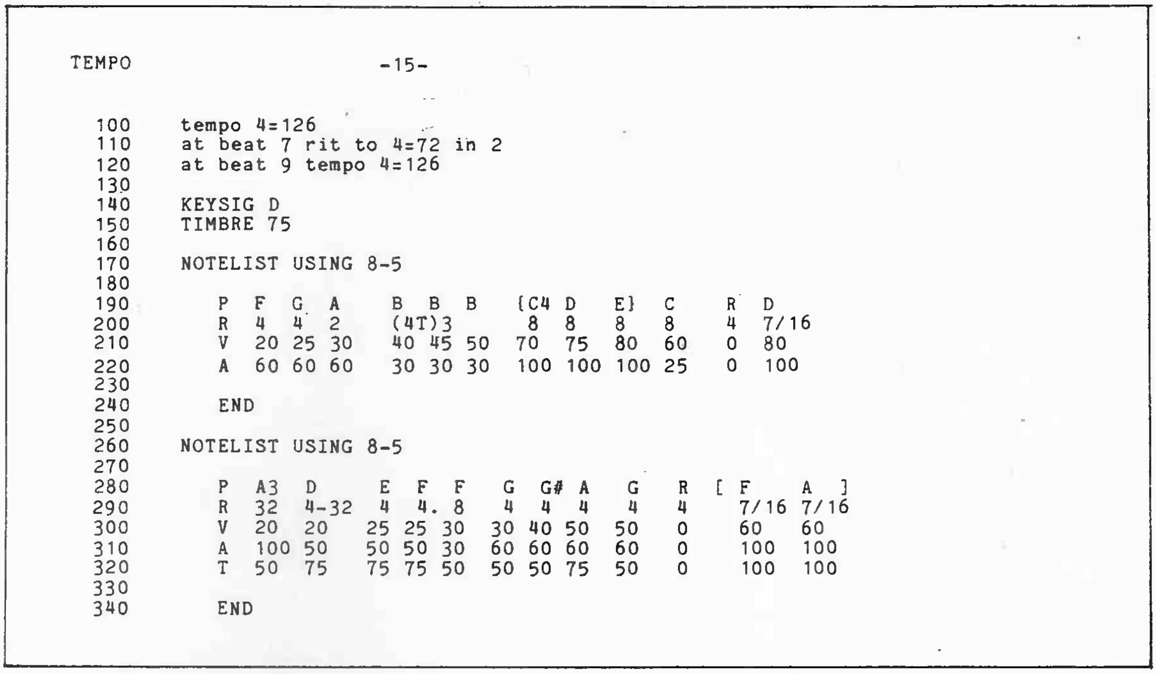
SYNC1   -13-   SCRIPT User Guide Appendix

```
 100    tempo 4=126
 110    at beat 7 ritard to 4=72 in 2
 120    at beat 9 tempo 4=126
 130
 140    fps 24
 150    delay beat 1 to :12
 160    sync beat 7 to :3:12
 170
 180    KEYSIG D
 190    TIMBRE 75
 200
 210    NOTELIST USING 8-5
 220
 230       P  F  G  A    B  B  B    {C4 D   E}  C    R  D
 240       R  4  4  2    (4T)3       8  8   8   8    4  7/16
 250       V  20 25 30   40 45 50   70  75  80  60   0  80
 260       A  60 60 60   30 30 30   100 100 100 25   0  100
 270
 280       END
 290
 300    NOTELIST USING 8-5
 310
 320       P  A3  D     E  F  F    G  G# A   G    R  [ F    A  ]
 330       R  32  4-32  4  4. 8    4  4  4   4    4    7/16 7/16
 340       V  20  20    25 25 30   30 40 50  50   0    60   60
 350       A  100 50    50 50 30   60 60 60  60   0    100  100
 360       T  50  75    75 75 50   50 50 75  50   0    100  100
 370
 380       END
```

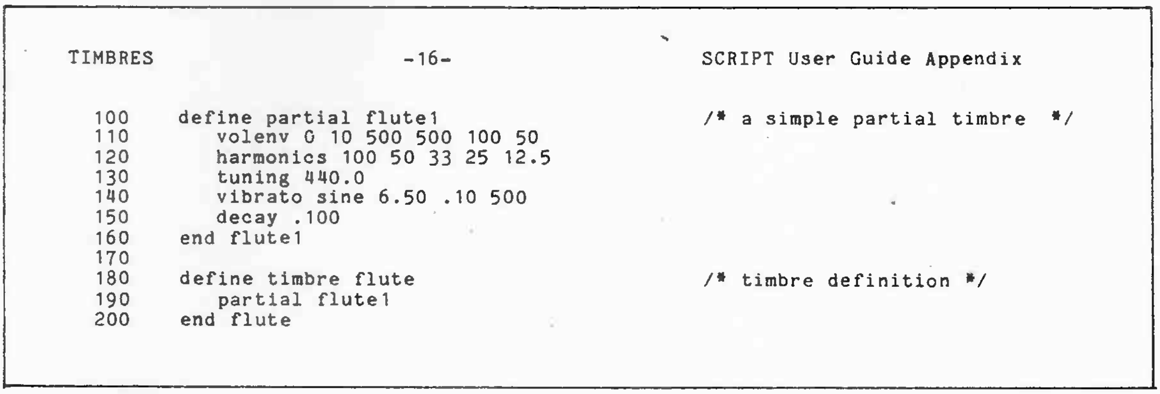


SYNC2   -14-   SCRIPT User Guide Appendix

```
 100    delay 4 to 4.5
 110    sync  8 to 8.5
 120    sync 12 to 12
 130    sync 16 to 16
 140    smooth 20 to 19
 150
 160    art mul 90
 170
 180    insert 'bach13'
```



```
TEMPO                       -15-

 100    tempo 4=126
 110    at beat 7 rit to 4=72 in 2
 120    at beat 9 tempo 4=126
 130
 140    KEYSIG D
 150    TIMBRE 75
 160
 170    NOTELIST USING 8-5
 180
 190       P  F  G  A    B  B  B   {C4 D   E}  C    R  D
 200       R  4  4  2    (4T)3      8  8   8   8    4  7/16
 210       V  20 25 30   40 45 50  70  75  80  60   0  80
 220       A  60 60 60   30 30 30  100 100 100 25   0  100
 230
 240       END
 250
 260    NOTELIST USING 8-5
 270
 280       P  A3  D     E  F  F   G  G# A   G   R  [ F    A  ]
 290       R  32  4-32  4  4. 8   4  4  4   4   4    7/16 7/16
 300       V  20  20   25 25 30  30 40 50  50    0    60   60
 310       A  100 50   50 50 30  60 60 60  60    0    100  100
 320       T  50  75   75 75 50  50 50 75  50    0    100  100
 330
 340       END
```



```
TIMBRES                       -16-                        SCRIPT User Guide Appendix

 100    define partial flute1                      /* a simple partial timbre  */
 110       volenv 0 10 500 500 100 50
 120       harmonics 100 50 33 25 12.5
 130       tuning 440.0
 140       vibrato sine 6.50 .10 500
 150       decay .100
 160    end flute1
 170
 180    define timbre flute                        /* timbre definition */
 190       partial flute1
 200    end flute
```