



Catalog Number 26-1922

# TRS-80® *Orchestra-90*

STEREO MUSIC SYNTHESIZER WITH PERCUSSION FOR MODELS III and 4

*SPECIAL COMPOSERS EDITION*



Model III or Model 4 with 16K Model III BASIC and Stereo System Required

## **The FCC Wants You to Know. . .**

This equipment generates and uses radio frequency energy. If not installed and used properly, that is, in strict accordance with the manufacturer's instructions, it may cause interference to radio and television reception.

It has been type tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna
- Relocate the computer with respect to the receiver
- Move the computer away from the receiver
- Plug the computer into a different outlet so that computer and receiver are on different branch circuits.

If necessary, you should consult the dealer or an experienced radio/television technician for additional suggestions. You may find the following booklet prepared by the Federal Communications Commission helpful: "How to Identify and Resolve Radio-TV Interference Problems".

This booklet is available from the US Government Printing Office, Washington, DC 20402, Stock No. 004-000-00345-4.

TERMS AND CONDITIONS OF SALE AND LICENSE OF RADIO SHACK COMPUTER EQUIPMENT AND SOFTWARE  
PURCHASED FROM A RADIO SHACK COMPANY-OWNED COMPUTER CENTER, RETAIL STORE OR FROM A  
RADIO SHACK FRANCHISEE OR DEALER AT ITS AUTHORIZED LOCATION

## LIMITED WARRANTY

### I. CUSTOMER OBLIGATIONS

- A. CUSTOMER assumes full responsibility that this Radio Shack computer hardware purchased (the "Equipment"), and any copies of Radio Shack software included with the Equipment or licensed separately (the "Software") meets the specifications, capacity, capabilities, versatility, and other requirements of CUSTOMER.
- B. CUSTOMER assumes full responsibility for the condition and effectiveness of the operating environment in which the Equipment and Software are to function, and for its installation.

### II. RADIO SHACK LIMITED WARRANTIES AND CONDITIONS OF SALE

- A. For a period of ninety (90) calendar days from the date of the Radio Shack sales document received upon purchase of the Equipment, RADIO SHACK warrants to the original CUSTOMER that the Equipment and the medium upon which the Software is stored is free from manufacturing defects. THIS WARRANTY IS ONLY APPLICABLE TO PURCHASES OF RADIO SHACK EQUIPMENT BY THE ORIGINAL CUSTOMER FROM RADIO SHACK COMPANY-OWNED COMPUTER CENTERS, RETAIL STORES AND FROM RADIO SHACK FRANCHISEES AND DEALERS AT ITS AUTHORIZED LOCATION. The warranty is void if the Equipment's case or cabinet has been opened, or if the Equipment or Software has been subjected to improper or abnormal use. If a manufacturing defect is discovered during the stated warranty period, the defective Equipment must be returned to a Radio Shack Computer Center, a Radio Shack retail store, participating Radio Shack franchisee or Radio Shack dealer for repair, along with a copy of the sales document or lease agreement. The original CUSTOMER'S sole and exclusive remedy in the event of a defect is limited to the correction of the defect by repair, replacement, or refund of the purchase price, at RADIO SHACK'S election and sole expense. RADIO SHACK has no obligation to replace or repair expendable items.
- B. RADIO SHACK makes no warranty as to the design, capability, capacity, or suitability for use of the Software, except as provided in this paragraph. Software is licensed on an "AS IS" basis, without warranty. The original CUSTOMER'S exclusive remedy, in the event of a Software manufacturing defect, is its repair or replacement within thirty (30) calendar days of the date of the Radio Shack sales document received upon license of the Software. The defective Software shall be returned to a Radio Shack Computer Center, a Radio Shack retail store, participating Radio Shack franchisee or Radio Shack dealer along with the sales document.
- C. Except as provided herein no employee, agent, franchisee, dealer or other person is authorized to give any warranties of any nature on behalf of RADIO SHACK.
- D. Except as provided herein, **RADIO SHACK MAKES NO WARRANTIES, INCLUDING WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.**
- E. Some states do not allow limitations on how long an implied warranty lasts, so the above limitation(s) may not apply to CUSTOMER.

### III. LIMITATION OF LIABILITY

- A. EXCEPT AS PROVIDED HEREIN, RADIO SHACK SHALL HAVE NO LIABILITY OR RESPONSIBILITY TO CUSTOMER OR ANY OTHER PERSON OR ENTITY WITH RESPECT TO ANY LIABILITY, LOSS OR DAMAGE CAUSED OR ALLEGED TO BE CAUSED DIRECTLY OR INDIRECTLY BY "EQUIPMENT" OR "SOFTWARE" SOLD, LEASED, LICENSED OR FURNISHED BY RADIO SHACK, INCLUDING, BUT NOT LIMITED TO, ANY INTERRUPTION OF SERVICE, LOSS OF BUSINESS OR ANTICIPATORY PROFITS OR CONSEQUENTIAL DAMAGES RESULTING FROM THE USE OR OPERATION OF THE "EQUIPMENT" OR "SOFTWARE". IN NO EVENT SHALL RADIO SHACK BE LIABLE FOR LOSS OF PROFITS, OR ANY INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY BREACH OF THIS WARRANTY OR IN ANY MANNER ARISING OUT OF OR CONNECTED WITH THE SALE, LEASE, LICENSE, USE OR ANTICIPATED USE OF THE "EQUIPMENT" OR "SOFTWARE".

NOTWITHSTANDING THE ABOVE LIMITATIONS AND WARRANTIES, RADIO SHACK'S LIABILITY HEREUNDER FOR DAMAGES INCURRED BY CUSTOMER OR OTHERS SHALL NOT EXCEED THE AMOUNT PAID BY CUSTOMER FOR THE PARTICULAR "EQUIPMENT" OR "SOFTWARE" INVOLVED.

- B. RADIO SHACK shall not be liable for any damages caused by delay in delivering or furnishing Equipment and/or Software.
- C. No action arising out of any claimed breach of this Warranty or transactions under this Warranty may be brought more than two (2) years after the cause of action has accrued or more than four (4) years after the date of the Radio Shack sales document for the Equipment or Software, whichever first occurs.
- D. Some states do not allow the limitation or exclusion of incidental or consequential damages, so the above limitation(s) or exclusion(s) may not apply to CUSTOMER.

### IV. RADIO SHACK SOFTWARE LICENSE

RADIO SHACK grants to CUSTOMER a non-exclusive, paid-up license to use the RADIO SHACK Software on **one** computer, subject to the following provisions:

- A. Except as otherwise provided in this Software License, applicable copyright laws shall apply to the Software.
- B. Title to the medium on which the Software is recorded (cassette and/or diskette) or stored (ROM) is transferred to CUSTOMER, but not title to the Software.
- C. CUSTOMER may use Software on one host computer and access that Software through one or more terminals if the Software permits this function.
- D. CUSTOMER shall not use, make, manufacture, or reproduce copies of Software except for use on **one** computer and as is specifically provided in this Software License. Customer is expressly prohibited from disassembling the Software.
- E. CUSTOMER is permitted to make additional copies of the Software **only** for backup or archival purposes or if additional copies are required in the operation of **one** computer with the Software, but only to the extent the Software allows a backup copy to be made. However, for TRSDOS Software, CUSTOMER is permitted to make a limited number of additional copies for CUSTOMER'S own use.
- F. CUSTOMER may resell or distribute unmodified copies of the Software provided CUSTOMER has purchased one copy of the Software for each one sold or distributed. The provisions of this Software License shall also be applicable to third parties receiving copies of the Software from CUSTOMER.
- G. All copyright notices shall be retained on all copies of the Software.

### V. APPLICABILITY OF WARRANTY

- A. The terms and conditions of this Warranty are applicable as between RADIO SHACK and CUSTOMER to either a sale of the Equipment and/or Software License to CUSTOMER or to a transaction whereby RADIO SHACK sells or conveys such Equipment to a third party for lease to CUSTOMER.
- B. The limitations of liability and Warranty provisions herein shall inure to the benefit of RADIO SHACK, the author, owner and/or licensor of the Software and any manufacturer of the Equipment sold by RADIO SHACK.

### VI. STATE LAW RIGHTS

The warranties granted herein give the **original** CUSTOMER specific legal rights, and the **original** CUSTOMER may have other rights which vary from state to state.



**TRS-80<sup>®</sup> Orchestra-90<sup>™</sup>**  
**Stereo Music Synthesizer**  
**with Percussion**  
**for TRS-80 III/4**  
**Special Composer's Edition**

**Radio Shack**  
A DIVISION OF TANDY CORPORATION  
FORT WORTH, TEXAS 76102

Orchestra-90™ Software Affair, LTD.  
© 1981, 1982, 1983 Jon Bokelman.  
Licensed to Tandy Corporation.  
All Rights Reserved.

All portions of this software are copyrighted and are the proprietary and trade secret information of Tandy Corporation and/or its licensor. Use, reproduction or publication of any portion of this material without the prior written authorization by Tandy Corporation is strictly prohibited.

Orchestra-90™ Owner's Manual  
Software Affair, LTD.  
© 1983, Jon Bokelman.  
Licensed to Tandy Corporation.  
All Rights Reserved.

Reproduction or use, without express written permission from Tandy Corporation, of any portion of this manual is prohibited. While reasonable efforts have been taken in the preparation of this manual to assure its accuracy, Tandy Corporation assumes no liability resulting from any errors or omission in this manual, or from the use of the information contained herein.

10 9 8 7 6 5 4 3 2 1

# Table of Contents

<b>Introduction.....</b>	<b>1</b>
System Requirements .....	1
<i>Using this Manual</i> . . . ..	2
Overview of the Orchestra-90 Package .....	2
<b>1/ Connecting the Orchestra-90 interface.....</b>	<b>3</b>
<b>2/ Powering-up the System.....</b>	<b>5</b>
Loading the Program (ORC90T).....	5
The Orchestra-90 Music Composer.....	7
A Mini-Concert .....	7
<b>3/ Creating a Sample File .....</b>	<b>9</b>
Scoring TTLS.....	10
Playing the Music File.....	10
Writing the Music File (Saving the File).....	10
An In-Depth Sample Session.....	10
<b>4/ Using the Orchestra-90 Programming System.....</b>	<b>19</b>
System Commands .....	19
Special Command Line Functions.....	23
Special Notes about Music Programming.....	23
Orchestra-90 Note Conversion Chart.....	25
Key Functions and Commands.....	27
Music Language Programming Functions.....	28
Articulations .....	28
Flats, Sharps, Accidentals .....	29
<b>5/ Transcribing Sheet Music .....</b>	<b>33</b>
<b>6/ Special Instrumentation .....</b>	<b>39</b>
Changing Instrumentation (Tone Color Registers).....	39
Percussion Instrumentation .....	40
Stereo Mapping .....	42
<b>7/ Music File Processing (ORCHUTIL) .....</b>	<b>43</b>
<b>8/ Technical Information .....</b>	<b>47</b>
Theory of Operation .....	47
Compiler Organization (SCORE).....	47
Parameters.....	49
<b>9/ Converting from Cassette to Disk.....</b>	<b>51</b>
Transferring Music Files on Cassette To Diskette (Disk Systems Only) .....	52
<b>10/ Configuring the Software for a Model III 4 MHz System (not for Model 4 users).....</b>	<b>55</b>
<b>11/ Error Messages .....</b>	<b>59</b>
<b>12/ Tempo Conversion Chart.....</b>	<b>61</b>
<b>13/ Miscellaneous Programming Examples.....</b>	<b>63</b>

Percussion Samples .....	65
<b>14/ Using Orchestra-90 with CompuServe.....</b>	<b>67</b>
Accessing and Using CompuServe's Orchestra-90 SIG.....	67
Downloading Music Files.....	67
Converting ASCII Music Files .....	67
<b>Appendix — Command Summary .....</b>	<b>69</b>
NOTE SIGNS.....	72
PERFORMANCE SIGNS .....	72



# Introduction

Radio Shack is proud to present the Orchestra-90™ package, which lets you become maestro to your own electronic orchestra. The Orchestra-90™ package is a unique combination of hardware and software that can transform any 16K TRS-80 Model III or Model 4 microcomputer into a programmable music synthesizer. Orchestra-90 does not make your computer a sounding keyboard.

The Orchestra-90 package features:

- *The Interface that connects your TRS-80's I/O Connector to your stereo to produce a stereo synthesizer.*
- *Two shielded phono cables for Stereo-to-Orchestra-90 connection. Please note that two Torroid rings are included in the package. Torroid rings are used to prevent interference to or from the Orchestra-90 hardware. Instructions on how they are used are given in the manual.*
- *Six octave ranges and up to five simultaneous notes or voices using two stereo channels.*
- *Easy to learn Music Language programming method which lets you enter all components line-by-line to make music.*
- *3, 4, or 5 voices. Each voice can be assigned to any of five instrumental choices (tone color registers). Experiment to create strange electronic sounds.*

The Orchestra-90 software consists of two programs which allow you to create, play, load, edit, and save music.

The Orchestra-90 Music Language was designed to let you transcribe virtually any written sheet music into a form that a computer can read and “play.” Musicians and non-musicians should find the language simple to learn and easy to use. The language has the features and capabilities to produce highly sophisticated computerized music.

The Music Language Program makes it possible to transcribe or compose music in any key or time signature. Diverse rhythm patterns are possible since the note values available range from whole notes to sixty-fourth notes. Notes may be single-, double-, or triple-dotted and/or played as triplets. Single and double accidentals are supported, as are articulations such as portato and staccato. You can also repeat sections of music, produce second endings (with or without retard) and change keys within a music file (modulation).

Once you have learned to program and save music files, you will want to examine ORCHUTIL. ORCHUTIL is a utility program for tape/disk transfer, music file packing, ASCII/Binary conversion, and Model I/III tape conversion.

## System Requirements

The minimum system required to run Orchestra-90 is:

- A 16K Model III Level II or a Model 4.
- A stereo.

## *Using this Manual . . .*

Years of musical training will certainly give you an advantage when transcribing music using Orchestra-90. However, enough background information is given in this manual to help you get started on your composing career. If you plan to create original music, a piano, guitar, or other instrument would be helpful to have at your disposal so that you can hear in advance what you are composing.

If you have a disk system, you will probably want to examine Chapter 9 for information on transferring tape programs and music files to disk.

After loading the programs, you will be shown how to load a music file. If you have properly set up the system, you will then be able to hear a sample of Orchestra-90's sound capabilities. Following the Orchestra-90 recital, a brief, yet memory-refreshing music theory review is given along with an explanation of how to transcribe sheet music using Orchestra-90's Music Language.

Composing and transcribing music files is no easy task at the beginning. The Music Language you use to program Orchestra-90 consists of many codes for traditional musical symbols. In this manual, all System Commands and Music Language Composition Codes are explained with examples. The process of creating music files using the Composition Codes is discussed followed by notes on saving, loading, and editing music files.

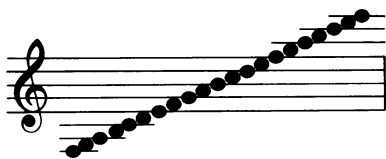
It is recommended that you begin transcribing some simple sheet music. Later, after you become comfortable with the music language, proceed to more sophisticated music. As with any other music instrument, practice is essential to becoming a virtuoso performer.

## **Overview of the Orchestra-90 Package**

Composing and arranging music with Orchestra-90 can be simplified by dividing the process into steps.

- Step #1.** Connect your computer to the Orchestra-90 Interface. Step #2. Connect the Interface Unit to your stereo.
- Step #3.** Power-up all equipment and load Orchestra-90 software. (Disk system owners can transfer the program from Tape-to-Disk.)
- Step #4.** Start Orchestra-90.
- Step #5.** Enter Orchestra-90's Edit Mode and use the Music Language to compose or arrange music.
- Step #6.** Return to the Command Mode and "SCORE" then "PLAY" the music.
- Step #7.** When you have programmed and arranged the music to your satisfaction, you can write the music file to cassette or disk for later listening pleasure.
- Step #8.** Use ORCUTL, the utility program, to manipulate music files.

When you have mastered the fundamentals of Orchestra-90, you can experiment with its advanced features. Discover the potential of Orchestra-90 through the guides to special instrumentation, percussion, and stereo mapping. Stereo mapping lets you direct certain voices from the right or left channels of the stereo.



## 1/ Connecting the Orchestra-90 interface

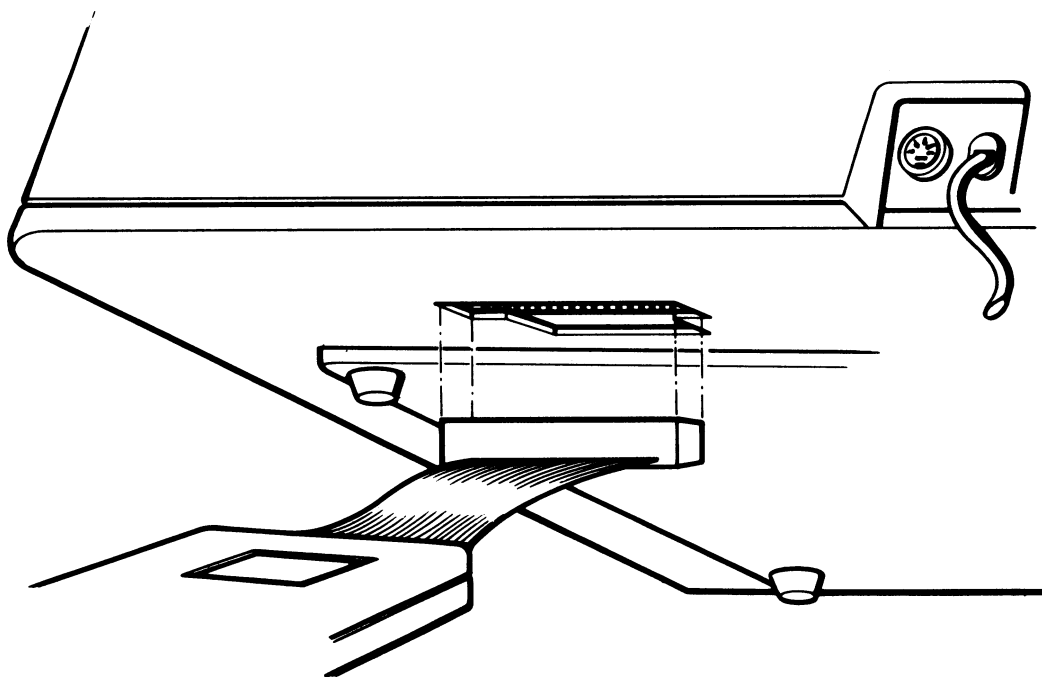
The Orchestra-90 Interface Module does not interfere with normal computer operations and can remain connected unless the I/O connector is required for another peripheral.

Always keep the amplifier at minimum volume when music is not being played.

**WARNING:** The Orchestra-90 Interface contains sensitive devices that can be damaged by careless handling. Always **TURN** the computer and amplifier **OFF** before connecting or removing the Interface. Also, it is recommended that you avoid using non-U.L. approved amplifiers.

Follow these steps to connect the Orchestra-90 Interface:

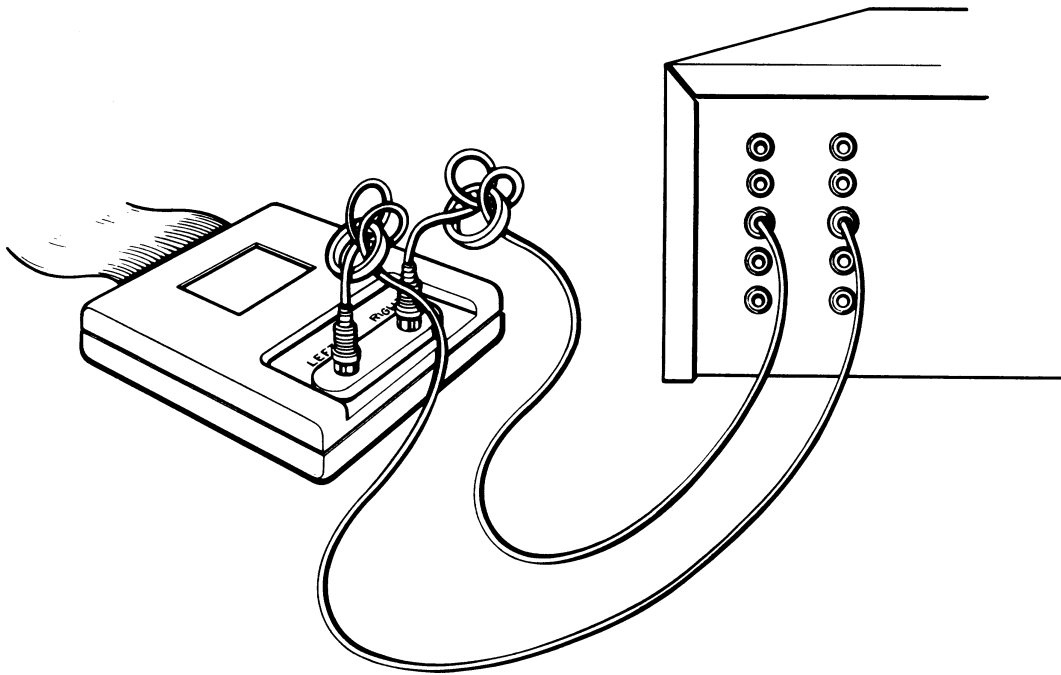
1. Plug the free end of the flat cable into the 50-pin I/O expansion connector located on the rear bottom of the computer. The Interface's flat cable and case should extend beyond the back of the computer.



**Figure 1. Installing the Orchestra-90 interface**

2. Take one of the Torroid rings included with the package and wind one of the phono cables through the ring three times, making sure that the Torroid ring is close to one end of the cable. Repeat this process for the other cable and Torroid ring.

The audio sockets on the box will mate with standard (RCA jacks) phono plugs. Connect the Torroid ring end of the phono cable to the phono socket on the box. Connect the other end to the **AUX**, **TAPE**, **TUNER** or other high level inputs of a stereo amplifier or receiver. **DO NOT** use the **PHONO**, **MIC**, or other low-level inputs.

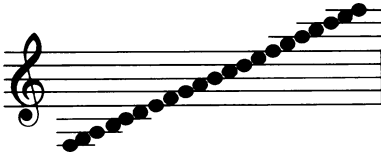


**Figure 2. Orchestra-90 Interface to Stereo Connection**

**Note:** Make sure that your stereo and computer are properly grounded.

3. After all connections are secure, turn the computer and the amplifier on.

Now the software can be loaded.



## 2/ Powering-up the System

This section explains system power-up, software configuration, and program initiation.

The software included in this package is on cassette tape. There are two versions of Orchestra-90 on this tape. Note that one side of the cassette is labeled **ORCH90** and the other side is labeled **ORC90T**. If you do not have disk drives, you will have to use the tape version of Orchestra-90 (**ORC90T**) to run the program. If you have a disk system, you can save the disk version (**ORCH90**) onto diskette. (Refer to Chapter 9 if you have a disk system.)

During configuration, you select the “clock rate” and the number of voices you want the synthesizer to support. The answers to these questions will depend on your computer’s MHz status. If your TRS-80 Model III has been modified for 4 MHz operation, you may instruct the program to take advantage of the faster clock rate and the five-voice option. Do not select the five-voice option unless you have a 4 MHz modified computer.

Orchestra-90 will automatically take advantage of the Model 4’s 4 MHz clock and no special clock rate configuration questions will appear. Orchestra-90 can play five-voice harmony with a standard Model 4 system.

The configuration procedure consists of a programmed sequence of steps. At each step the program will ask a question (a system prompt) and wait for your answer. In most cases, the range of acceptable answers will be displayed with each question. After typing your response, press **ENTER**. If your answer is accepted, the program will proceed to the next step. If not, the question will be repeated until an acceptable answer is obtained. Your answer should not contain any spaces. Pressing the **BREAK** key at any time will clear the entry and let you enter another answer.

Note: If you have a TRS-80 Model III modified for 4 MHz operation, please turn to Chapter 10 where you will be given specific configuration instructions. These special instructions are not applicable to Model 4 owners.

If you have not already done so, connect the Orchestra-90 interface board to your computer.

### Loading the Program (ORC90T)

1. Before powering up the system, make sure that your cassette recorder is properly connected to your computer. Insert the tape (label-up) that contains the **ORC90T** program.
2. Set the cassette recorder to the PLAY mode. Press the orange **RESET** button.
3. The prompt displays:  
CASS?  
In response:
  - Press **H** for high speed.
4. The prompt displays:  
MEMORY SIZE?  
  - Press **ENTER**
5. The prompt displays:  
Radio Shack Model III basic  
READY  
>

- Type: **SYSTEM** **[ENTER]**
6. The prompt displays:
- \*?
- Type **ORC90T** **[ENTER]**
- There will be a short pause while the program loads. During that time, the cursor will not blink. There will be a blinking asterisk in the upper left corner of the screen. Once the program is loaded, another prompt will appear.
7. The prompt displays:
- \*?
- Type / **[ENTER]** and the program will begin.
- When the program starts and the screen clears, a copyright notice will appear at the bottom of the screen. The first question of the configuration dialog will appear in the middle of the screen with a blinking cursor on a line below it.
8. The following prompt appears:
- DUPLICATE?<Y/N>
- At this point, you have the option to make a backup (copy) of the tape program. It is recommended that you make a backup now, and use the backup to run the program. Keep the original in a safe place.
- To proceed and ignore the Backup option, type **N** **[ENTER]**.
  - Before answering Y, make sure that you have a blank cassette in the cassette recorder and the recorder is in the RECORD mode. The program will begin Writing when you press **[ENTER]**.
  - If you have a Model 4, proceed to Step 10.
9. The prompt displays:
- FAST CLOCK? < Y/N>
- Type **N** **[ENTER]** if your system is not equipped for 4 MHz.
10. The next prompt displays:
- HOW MANY VOICES? <3/4/5>
- This is asking for the number of voices you want the synthesizer to have. Or in other words, what is the maximum number of parts you want to hear at any given time.
- You can select 3 or 4 voices if your system is a standard Model III without a 4 MHz modification. If you have a Model 4, you can select 3, 4, or 5 voices. However, since we are about to play the sample music files which are arranged in 4 voices, select the 4 voice option for optimum sound quality.
11. The last prompt in the series displays:
- SAVE PROGRAM? <Y/N>
- Type **Y** **[ENTER]** if you want to save the configuration as you just entered it. The program will be permanently configured and cannot be changed.
  - Type **N** **[ENTER]** to not save the configuration as entered.
- The cursor will then move to the upper left corner of the screen.

# The Orchestra-90 Music Composer

The screen shows:



- The Command Line is where you initiate all system commands. For example, you can **PLAY** a song or **WRITE** a song to tape using this line.
- The Programming Line is where you will be programming and editing music using the Music Language.

All Commands and Music Language Codes will be thoroughly explored in this manual,

## A Mini-Concert

Now that the program (**ORCH90** or **ORC90T**) is loaded, it is time that you had a sample of how previously programmed music sounds through your stereo. There are sample music files included on the Orchestra-90 cassette tape. These files are **GYPSY**, **ENTRY**, **CZAR**, and **RACES**.

To actually hear what Orchestra-90 can do, make sure your stereo is turned to a low volume and that all equipment is connected correctly. Make sure that you have the correct side of the tape in the cassette player and **REWIND** the tape. Then, set your cassette recorder to the **PLAY** mode. At the Orchestra-90 Command Line, type:

**GET GYPSY** **ENTER**

The cassette will start. There will be a pause while the system searches the cassette for **GYPSY**. Once the music file has been located, if the stereo is correctly connected, you should hear **GYPSY**, a classical rondo by Joseph Hadyn. Just sit back and enjoy. If you do not like classical music, press **Ø** and the music stops. The file remains in memory and may be played again by typing **P** **ENTER**.

The piece will play about five minutes. After completion, or if you press **Ø**, the cursor at the Command Line will be blinking. From the Command Line, you can play the other songs included on the tape by typing **GET ENTRY**, **GET CZAR**, or **GET RACES** (pressing **ENTER** after each command).

There are two ways to move from the Command Line to the Music Programming Line. You can type:

**E** **ENTER** (where **E** signifies **EDIT**)

or, you can type:

**SHIFT** **BREAK**

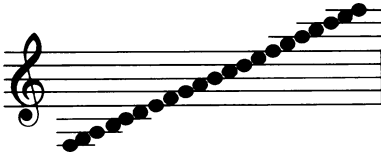
To move to the Programming Line, type **E** **ENTER**

The cursor will be blinking at the center of the screen at the first line of GYPSY. Use the **↓** and **↑** keys to scroll through the lines of the GYPSY program. If you press a key, it will repeat until you release it.

As you can probably tell, programming a five minute song requires a great deal of patience — especially in the beginning when you are learning the Music Language codes. The following sample session was developed to make a novice Music Language programmer feel more comfortable with Orchestra-90 programming methods.

Press **BREAK** to return to the Command Line.





### 3/ Creating a Sample File

This section of the manual is designed to help you feel comfortable with Orchestra-90 and give you an idea of what it can do. First, you'll learn how to create a music file. Then, you'll be told how to play and save that file.

#### *A Reminder to Orchestra-90 Composers...*

As with any program in which you create data files, you will want to regularly make backup copies (tape or disk) of your music files.

Your Orchestra-90 program (4-voice) should be loaded and configured according to the instructions previously explained in the manual. A blinking cursor should be displayed in the upper left corner of the screen, and if you have just played **GYPSY**, the code should be displayed on your screen.

To clear the Programming Line without affecting the Orchestra-90 program, type **N** **ENTER** (where N stands for NEW). You can load **GYPSY** later for another concert.

To move to the Programming Line and enter music, type:

**E** **ENTER**

or

**SHIFT** **BREAK**

You now have a blinking cursor in the center left of the screen.

That familiar standby, *Twinkle Twinkle, Little Star (ITLS)*, can easily be programmed using the Music Language. This particular version of *ITLS* has no extras. Only one voice used — no chords, just the bare essentials. Later, you will encounter the musical condiments that will embellish your compositions or arrangements. But before you attempt to transcribe *J.S. Bach's Brandenburg Concertos*, type the following:

**NQ=60** **ENTER**

**M1 Q3,3,7,7,\$** **ENTER**

**M2 Q8,8,7,\$** **ENTER**

**M3 Q6&,6&,5,5,** **ENTER**

**M4 Q4,4,3,\$** **ENTER**

**M5 Q7,7,6&,6&,** **ENTER**

**M6 Q5,5,4,\$** **ENTER**

**M7 Q7,7,6&,6&,** **ENTER**

**M8 Q5,5,4,\$** **ENTER**

**M9 Q3,3,7,7,** **ENTER**

**M10 Q8,8,7,\$** **ENTER**

**M11 6&,6&,5,5,** **ENTER**

**M12 4,4,3,\$** **ENTER**

If you make a mistake while entering this information and cannot seem to correct the error using the trial and error technique, refer to the section entitled **Programming Functions**,

## Scoring TTLS

Before playing this piece, you must first return to the Command Line and **SCORE** TTLS. When you **SCORE** a music file the system will check for errors and compile (translate into machine language) the piece. Remember, you cannot **PLAY** a newly programmed piece or a newly edited piece until it has been **SCORED**. To return to the Command Line, press **BREAK**.

To **SCORE** the piece, type:

**S** **ENTER**

After a brief pause, the piece will be scored. If there is a mistake in the code, an error number will appear in the upper left corner of the screen. The actual error in the code as detected by the system would be highlighted. Any time an error correction or change is made in a music file, the file must be re-**SCORED** before it can be played.

## Playing the Music File

After **SCORING** the music file, make sure that your stereo is properly connected to the Orchestra-90 Interface. Turn down the volume of the stereo (you can adjust it later). The command symbol used to **Play** the piece is **P**.

Type **P** **ENTER**

After the piece has played, you can listen to it again by typing **P** **ENTER**.

## Writing the Music File (Saving the File)

You can save the Music File onto cassette tape or disk to prove that you actually composed music. Make sure that you have a new cassette tape in the tape recorder and that the recorder is set to **RECORD**.

At the Command Line, type:

**W TTLS** **ENTER**

After you have saved TTLS, at the Command Line type **N** **ENTER** to clear the memory.

## An In-Depth Sample Session

This portion of the sample session will describe the process of transcribing traditional music symbols to Orchestra-90 codes. If you have had no prior musical education, keep in mind that there is a code for each note and a code for each value of a note.

To interpret sheet music correctly, it helps to know standard musical notation. Included in this section are several charts and examples designed to help explain a few problems encountered during Music Language programming.

First, you will enter the information contained in the musical passage in Figure 3. The traditional music symbols are explained along with the Music Language substitutes.

### Music sample #1

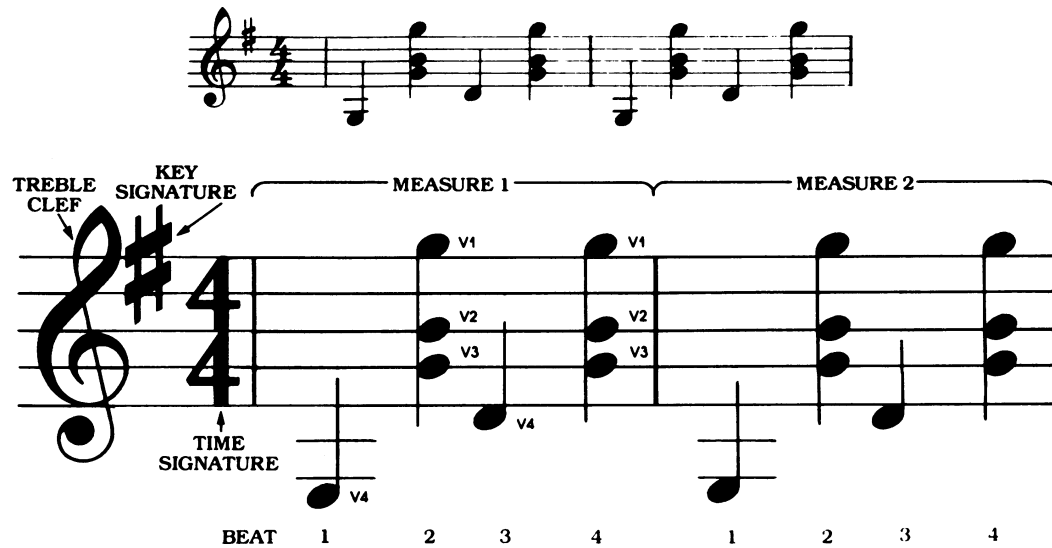


Figure 3.

The five, long, horizontal lines are known as Ledger Lines. Together they are known as the Staff. Most of the information you will be working with may be found on the staff or on additional ledger lines added above or below the staff for extra high or extra low notes.

Reading from the left side of the Ledger Line we first encounter the Treble Clef symbol. All of the music in the following example will be in the treble clef. The note symbols for the treble clef are found in Tables 1 and 2. The upper half of the diagram represents the Treble Clef, the bottom half is the Bass Clef.

To the immediate right of the treble clef is a single # mark known as a Sharp. This one Sharp tells a very important fact about the piece of music — in what key it is to be played. One Sharp denotes the key of G.

To tell the system to play in the key of G type:

K1# **ENTER**

If you have programming experience, you may be wondering where the line numbers are. There is no need for line numbers while programming with Orchestra-90.

The next piece of information could be placed on the same line as the key signature, but for the sake of clarity we will use separate lines for each new item.

Refer to Figure 3. The 4/4 on the staff is the Time Signature. In traditional sheet music, this consists of two numbers, one on top of the other, similar to a mathematical fraction.

For example, 4/4, 5/4, 3/4, 6/8, etc.

The top number tells us the number of Beats in a Measure. A Measure is a section of the staff between two vertical lines called Bar Lines which extend from the top line of the staff to the bottom line. This example has four beats per measure. You will get the feel of the music beat by tapping your foot and counting out loud 1 - 2 - 3 - 4.

Next, examine the bottom number in the time signature which is another 4. This 4 means that each beat in the music will be equal to a Quarter Note. Therefore, there are 4 beats in each measure and each one is equal to a quarter note. Refer to the chart in the Music Language Summary section and make sure you know what this note looks like. This example contains all quarter notes. This is called 4/4 Time or Common Time.

In Music Language programming, a line must be added to describe the Tempo, or how fast the music is to be played. This is indicated differently than the previous explanation of traditional sheet music notation.

Type:

NQ=80 **ENTER**

This line indicates that a note (quarter note) will be equal to a time of 80 (HEX). If you start out with this tempo value, you can always change it later to speed-up or slow-down the piece. The speed can also be determined while “playing” a song. See the Tempo Conversion Section.

So far, the Music File is:

K1#  
NQ=80

As already mentioned, you can program up to five voices per measure. By programming several voices per measure, you create a polyphonic sound. The example is suited for four Voices.

Orchestra-90 can be programmed to make each voice an individual sound. There are five standard sounds available — A, B, C, D, and E.

A = Trumpet  
B = Oboe  
C = Clarinet  
D = Organ  
E = Violin

To assign sounds A, B, C, D, to Voices 1, 2, 3, and 4 respectively, type:

V1YA V2YB V3YC V4YD **ENTER**

This line indicates that Voice 1 is to be set (Y) at Register A, or is to sound similar to a Trumpet. Voice 2 is set (Y) at Register B or is meant to mimic an oboe. Voice 3 is set (Y) at Register C and should sound like a clarinet. And, Voice 4 is set (Y) at Register D and sounds like an organ.

The music will begin with a Part Number which contains one or more measures and can be repeated. Measures cannot be repeated directly, only Parts containing measures. Type:

P01 **ENTER**

Parts may consist of any two-digit numbers, Be careful not to identify two different parts by the same number.

The next essential code is the Measure symbol. Type in the first measure symbol but don't press **ENTER** because the Measure symbol is followed by information on the same line.

M01

Now press the spacebar and type **\*** (without pressing **ENTER**). The asterisk indicates that all of the notes following are in Treble Clef.

M01 \*

Now the notes can be entered for the Voice 1 in Measure 1. The first line in a measure is implied V1 so you do not have to specify Voice 1 in the code. To indicate that the notes belong to a specific voice, a Voice code is entered, For example, V2.

In Figure 3, V1 has notes on beats 2 and 4 only. Beats 1 and 3 are silent and must be entered as rests. The Music Language's code for a Rest is the \$. Q\$ added to the line presents two new pieces of information. Q indicates that all notes or rests which follow have the time of a quarter-note. The \$ is the symbol for any rest. In this case, it follows a quarter note time value symbol, so the first beat becomes a quarter note rest.

Now the notes for Voice 1 can be added. Checking the scale in Table 1 or 2 in this manual, you'll find the symbol for this note to be a B. Add this to the line:

M01 \*Q\$B

V1 does not appear in the third beat of the measure, so another rest is added which makes the line:

M01 \*Q\$B\$

The note on beat four is also B. Type **B** and press **ENTER** to begin entering another line:

M01 \*Q\$B\$B **ENTER**

V2 can be defined the same way. The rests occur in the same places, the only difference being that the notes are the harmonizing 6 instead of B.

To enter the code for the Voice 2 of Measure 1, first press the spacebar so that the cursor is located directly beneath the \* in the previous line.

Then, type:

V2Q\$6\$6 **ENTER**

To recapitulate, the code for Measure 1 should now look like this:

M01 \*Q\$BSB V2Q\$6\$6

When entering the data for a Measure, you can separate each Voice line, or you can enter the Measure in the format:

M01 \*Q\$B\$B V2Q\$6\$6

Voice 3 has the same timing as the notes (4) harmonizing with B and 6. To enter V3. type:

M01 \*Q\$B\$B

V2Q\$6\$6

V3Q\$4\$4

You have defined 3 voices. When the music is played, the three voices played in unison make a chord.

Voice 4 has notes on beat 1 and 3. It has rests on beats 2 and 4. The first note is a -3. The note on beat 3 is a 1. Enter Voice 4 as shown:

M01 \*Q\$B\$B

V2Q\$6\$6

V3Q\$4\$4

V4Q-3\$1\$

The measure is finished. To listen to the music type:

**BREAK** (Press once to leave editor)

**S** **ENTER** (Score or Compile the Music)

**P** **ENTER** (You hear the finished music)

To return to the editor:

**E** **ENTER**

Measure 2 is an exact duplicate of Measure 1. To Repeat Measure 1 or Part 01. type:

R01 **ENTER**

That tells the system to repeat Part 01 (which contains measure 1). The example is now completely transcribed:

K1#  
 NQ=80  
 V1YA V2YB V3YC V4YD  
 P01  
 M01 \*Q\$B\$B  
     V2Q\$6\$6  
     V3Q\$4\$4  
     V4Q-3\$1\$  
 R01

Most sheet music will have the rests clearly written in. Correct transcription will depend on your awareness of timing. The number of beats per measure, the type of note that gets one beat, and the interpretation of rests all are factors that influence timing.

Music Sample Two gives you some practice with editing — changing the tempo, altering the voicing, and adding repeats. Score and play the music after each change.

### Music Sample #2

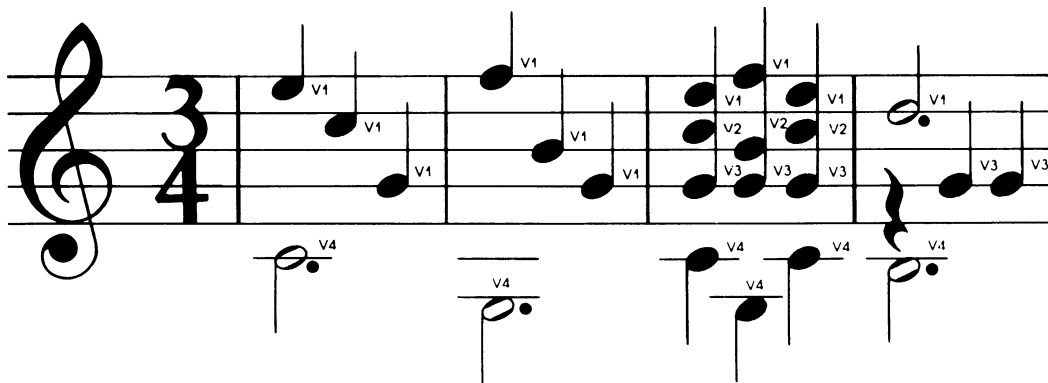


Figure 4.

This session describes and interprets the music shown in Figure 4. Study the sample music and the finished transcription. Remember this transcription is only one possible interpretation. The final decision should be based on what you believe sounds best.

Enter, **SCORE**, and **PLAY** this piece:

K0#  
 NQ=80  
 V1YA V2YB V3YC V4YD  
 P01  
 M01 \*Q974  
     V4H.0  
 M02 \*QA64  
     V4H.-3  
 M03 \*Q9A9  
     V2Q767  
     V3Q444  
     V4Q0-30  
 M04 \*H.8  
     V3Q\$44  
     V4H.-1

Notice that there are no sharps or flats in the key signature. Enter **K0#**. Figure 4's time signature tells you that there are three quarter-note beats in each measure.

Measure 1 requires two voices — V1 for the melody notes and V4 for the bass notes. The melody notes are usually the highest notes in the measure and it follows that the bass notes are the lowest. V2 and V3 are not used and therefore not defined in Measures 1 and 2.

Measure 1 contains a dotted half-note, H., followed by the note symbol for Middle C which is 0. A half-note lasts twice as long as one quarter-note. A dotted half-note has its value lengthened by 1/2. The total value of a dotted half-note is equal to three quarter-note beats. In 3/4 time, three beats fill the measure.

In Measure 2, the timing is the same, but there are different note symbols.

Measure 3 uses all voices and presents another four-part harmony using all four voices. All the notes are quarter-notes.

Measure 4 has three voices. V2 is ignored. V1 consists of a single dotted half-note. V3 has two quarter-notes played after a quarter-note rest(\$).

V3Q\$44

V4 is a dotted half-note — symbol -1.

Listen again to this piece, especially Measure 4.

The two notes in V3 have no articulation and sound like one continuous note. Change V3 in Measure 4 to:

V3Q\$4'4

The ' is an articulation mark which makes the notes sound more distinct.

To produce an effect called Staccato, add a semi-colon (;) behind each note in V1, Measures 1 and 2:

M01 \*Q9;7;4;

M02 \*QA;6;4;

SCORE and PLAY this new version. The traditional musical notation for staccato is a dot directly above or below the note.

Experiment with articulation and staccato to produce different effects in your music.

### Music Sample #3

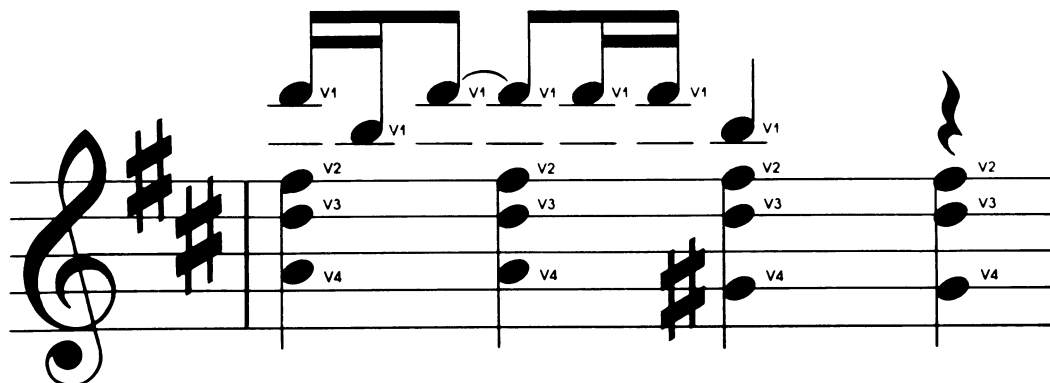


Figure 4A.

This measure is in 4/4 time. The example is extremely important. Enter the file exactly as shown:

```
K2#
NQ=80
V1YA V2YB V3YC V4YD
P01
M01 *SF'D' IFF;SF;F;QD$
    V2QAAAA
    V3Q8888
    V4Q554#4
```

**SCORE** and **PLAY** the piece. The melody notes (V1) are so high that they are barely audible. A change can be made that will bring the melody down to a comfortable frequency. This can be done two ways.

If we want to lower only V1, we can add a voice modifier after the part number.

```
P01 V1U-7
```

V1U-7 instructs the system to play the notes in V1 one octave lower than written. Make this change. **SCORE** and **PLAY** it. The melody is more audible; however, the other voices are still too high. A better way is to transpose the entire piece of music down several half-steps. This lowers all the voices equally.

Remove V1U-7 from the line and replace it with <9.

```
K2#
NQ=80
V1YA V2YB V3YC V4YD
P01<9
M01 *SF'D' IFF;SF;F;QD$
    V2QAAAA
    V3Q8888
    V4Q554#4
```

**SCORE** and **PLAY**. The entire piece has been lowered by 9 half-steps, or slightly over half an octave. Try replacing the 9 with an A, C, or any hex digit up to F. The tone quality of the voices improves and distortion of the high notes is reduced.

V1 starts out with two sixteenth-notes, F and D, each followed by an ' (articulation to make these notes more distinct).

The third and fourth notes are eighth-notes F. The curved line between the two notes indicates that the notes are Tied, meaning no articulation on the first F. The two notes must sound without interruption. The second F has articulation to separate it from the two sixteenth-notes that follow.

The two sixteenth-notes, also F's, have articulation because they are not tied.

A quarter-note D follows without articulation.

The last beat in V1 is a rest (\$).

Both V2 and V3 consist of four quarter-notes. V2 has four A notes and V3 has four 8 notes.

V4 also has four quarter-notes. The first two notes are 5, the third note has an accidental applied to it. In musical terms, a sharp preceding the note means that the note and all following notes of the same symbol are to be played 1/2 step higher than normal for the duration of that measure. Our system carries the accidental automatically so the first note is entered as 4# and the following note is entered just like it is written, 4. The second 4 will be played as a 4# no matter which voice plays it.



After playing the original transcription, change V2, V3, and V4.

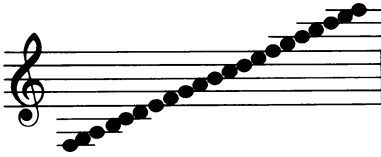
V2QA;A;A;A;

V3QB:8;8;8;

V4QS;5;4#;4;

A; has been added to every note. **SCORE** and **PLAY** this new version and you will hear a completely different musical effect. The chords have a shorter duration which tends to accent the four beats in the measure.





## 4/ Using the Orchestra-90 Programming System

This gives an in-depth description of System Commands and Music Programming Commands. Both types of commands are essential to creating music files. The following discussion will describe each command and give examples of practical uses. Some of these commands are ones which you experienced while studying Creating a Sample File in the previous section. In the Appendix you will find a handy quick reference section listing brief descriptions of commands and music symbols.

### System Commands

System Commands are entered using the top line of the Orchestra-90 screen — the Status Line. All system commands consist of a keyword (or keyword abbreviation) and are sometimes followed by one or more operands. Operands are separated from each other and from the keyword by one or more spaces. The keyword may be abbreviated (only the first character is significant) but may not contain any spaces. If you enter an unrecognized command, a ? is displayed in the status line.

**N**

*Clear the Music File Buffer*

**NEW**

Type **N** **ENTER** to delete the current file in memory and begin a new file.

**T**

*Go to the first line of the music file*

**TOP**

Type **T** **ENTER** to move the cursor to the top or beginning of the current file.

**B**

*Go to the last line of the music file*

**BOTTOM**

Type **B** **ENTER** to move the cursor to the bottom or end of the current file.

**E**

*Go to the music file programming line*

**EDIT**

Type **E** **ENTER** to enter the Edit (Programming) Mode.

**/**

*Go to the specified string*

**/string**

Use this command to search from the current cursor location in the music file for a specific phrase, word, or number. Type **/**, then the item you are searching for and press **ENTER** (Hold down the **ENTER** key for a continuous search.) If the string or item is located, the Programming Line's cursor will pinpoint it. The program returns to the System Command line. To do a search for the same item, simply press **ENTER**.

For example, to search a music file for a string such as P01, at the Command Line you would type:

**/P01** **ENTER**

### *Go to the string from the bottom*

#### **-string**

This command is similar to the /string command; however, this command searches from the current cursor location in the music file to the top. Both “/” and “-” are auto-repeat commands. That is, the Command Line is not cleared after execution. This allows you to search forward and/or backward for each occurrence of a string by holding down the **[ENTER]** key. After the entire file has been searched, you will need to re-specify the command again to start the search over.

For example, to search a file for a string such as M27, you would type:

**-M27 [ENTER]**

#### **R**

### *Load a file into memory*

#### **READ filename**

**Cassette Users:** This command will load a music file into memory but will not “play” the file when it is located. To READ a file, at the System Command line type:

**R filename [ENTER]**

The routine searches the tape for a matching filename. When found, the file is read into memory replacing any previous file there. An asterisk (\*) in the file name will match any character. The command R xx\* will read the next file with a name beginning with xx. During the search, the name of each file encountered will be displayed in the status line. The operation may be terminated at any time by holding down the **[BREAK]** key. (The tape must be non-blank and moving.)

**Disk Users:** The routine searches the disks for a matching filename. When found, the file is read into memory replacing any previous file there. If not found, an error code is displayed in status line. The filename may be followed by a drive number. The file extension is always **/ORC**.

#### **W**

### *Save a file by the name assigned*

#### **WRITE filename**

A file can be saved to tape or diskette at any time, regardless of whether you have SCORED the music file or not. The procedure is somewhat different for Cassette Users and Disk Users.

**Cassette Users:** This command is used to save music files to cassette tape for later listening pleasure or editing. When using this command, make sure that you insert a new cassette tape (not the program tape) into the cassette recorder. Set the recorder to RECORD. Type **W**, an original filename for the music file, and press **[ENTER]**.

For example, to save a file called **MOZART**, at the System Command Line you would type:

**W MOZART [ENTER]**

**Disk Users:** This command is used to save music files to diskette. To save a music file to diskette, type an original filename, a colon and drive number if you have more than one disk drive, and press **[ENTER]**. The current file is written with the filename given. The system automatically adds the **/ORC** file extension.

For example, to save a file called **ALLEGRO** to Drive 2 of your system, type:

**W ALLEGRO:2 [ENTER]**

The cursor will resume flashing after the file has been saved.

**V** (for cassette users)

### *Verify that file is written to tape*

#### **VERIFY filename**

After **WRITING** a music file to cassette using the **W** command (the file is still in memory), you can verify that the file has been saved by typing **V**, the filename, and **[ENTER]**. If the file has not been saved correctly, an error code will be displayed. If there is an error code, repeat the **WRITE** procedure and **VERIFY** again.

For example, to verify the saved file, **ALLEGRO**, type:

**V ALLEGRO [ENTER]**

**A**

**Add another file into memory**

**APPEND filename**

Append, or add, an existing music file to the beginning of a music file currently in memory using the **APPEND** command.

For example, if you had a file currently in memory, you could add another file from tape called **FORTE**. You would type:

**A FORTE [ENTER]**

After the appended file has been loaded, the cursor will resume blinking. You can append more files or edit the files in memory. You can also **SCORE** several files at once to produce a medley. (This may require some reassigning of part numbers.)

**D**

**Directory**

**DIR number**

If you have a disk system, you can verify the directory's contents by typing **D**, the drive number, and **[ENTER]**. This command only displays the names of the Orchestra-90 music files on the disk (files with **/ORC** extension).

For example to check drive **I**'s directory, type:

**D1 [ENTER]**

The directory for that disk will then appear on the screen, Press **[SHIFT] [BREAK]** if you wish to return to the Programming Line of a music file already in memory.

**K**

**Delete a music file**

**KILL filename**

You can eliminate a music file from your directory. To Kill a file, type:

**K filename [ENTER]**

Then, check your directory to make sure that the file has been deleted.

**Q**

**End session**

**QUIT**

After a few hours of composing or arranging music files, you may be a bit fatigued. To exit the Orchestra-90 program, type:

**Q [ENTER]**

When using cassette, the system will normally return to Level I Basic. If this does not happen, you will need to press the computer's Reset button.

When using a disk drive, the system will return to **TRSDOS Ready**.

## L

### *Printout a Music File*

#### LIST

If you have a printer, you can print out music files for future reference. Make sure your printer is correctly connected to your computer and On Line. Type:

L **ENTER**

The printer starts. To stop printing, you can press **BREAK**.

## S

### *Compile a Music File*

#### SCORE

Before you can listen to a music file, you must use this function to Score (compile) the current file into the binary (coded) form required by the synthesizer. A newly created music file may not be played until you have SCORED the file.

When using the Score command, the compiled binary code exists only in memory during play of a song. The compiled binary code cannot be saved to disk.

After finishing part or all of a music file, press **BREAK** to return to the Command Line. Then, to SCORE a music file, type:

S **ENTER**

You can return to the Command Line at any stage of composition, **SCORE** the music file, and then **PLAY** the file. Remember, a file cannot be PLAYED before it has been scored. During the scoring process, the system evaluates your file and if there is something wrong with it, such as a non-existent note, an error message will appear in the upper left portion of the screen. A white block will occur at the point of error. (Refer to Chapter 11 for a list of error messages.) After you have corrected the error, press **BREAK** to return to the Command Line and re-SCORE the piece.

If there is not enough memory to hold the music file and the compiled object code at the same time, the program will pause and display the question, OVERLAP? in the status line. If you answer Y, compilation will resume and portions of the source file will be overlapped by the object code. Answering anything else will cancel the compilation and enter the command mode.

The **OVERLAP** function is useful when compiling large files. The end of the object code can **OVERLAP** the beginning of the source code. DO NOT OVERLAP a file that has not been saved on disk or tape since this function implies the probable destruction of the current file. Do not expect to be able to use it in any way once this command is given.

## P

### *Play from the part number specified*

#### PLAY part number

This command tells the synthesizer to **PLAY** the most recently SCORED music file starting with PART nn. If nn is omitted, the entire piece is played (press **0** to stop). An error message will be displayed if the PART cannot be found or if the text file was not recently SCORED.

For example, to play from Part 03 to the end of a piece, at the Command Line you would type:

P 03 **ENTER**

## G

### *Read, Score, and Play the specified file(s)*

#### GET filename

This is a multifunction command that will perform a **READ**, **SCORE**, and **PLAY** for each of the file names. OVERLAP will be performed automatically, if necessary.

If you are a cassette user, please note that you must have the REMOTE plug connected to the recorder or the tape will continue running while the first file is playing. The GET command automatically switches the cassette player ON and OFF between songs only if the REMOTE plug is connected.

**Disk Users:** Filenames may be followed by a drive number just as in the READ command.

**Cassette Users:** Filenames may contain \* as in **READ**. In addition, an at-sign (@) will match any filename, repetitively. **GET @** will **READ**, **SCORE**, and **PLAY** all files on the tape.

**M** (for disk users only)

#### Multiple GET for perpetual play of specified files

**MULTI filename1 filename2 filename3 filename4 . . .**

This command is a perpetual **GET** and is useful in background music application. Each file will be loaded and played. After the last specified file has played, the system will automatically start the process again and play the same files again.

For example, to load and play three files repeatedly, you might type:

**M WALTZ SONATA POLKA**

## Special Command Line Functions

The following three commands will be especially useful after you begin programming music. Remember, these commands are entered at the Command Line.

**[SHIFT] [!]** This command lets you play from any point (cursor position) in the music file. At the Programming Line, use the arrow keys to move to the position of the piece you wish to hear. Press **[BREAK]** to return to the Command Line. The cursor will stay at the point in the piece where you exited. To hear the remainder of the piece from that cursor point, type **[SHIFT] [!] [ENTER]**. This command ignores all notes prior to the cursor location during compilation, therefore if the portion compiled contains a Repeat of measures in the section prior to the cursor location, then that Repeat will have no sound or duration.

**[@]** This command lets you pinpoint a wrong note or any specific passage in a piece upon “playing” the piece. When the piece is playing and you want to stop it type **[0]**. Then, at the Command Line, type **[@] [ENTER]** to return to the general area where you stopped the previous play. This command is useful for finding errors heard during Play. You will need to re-SCORE after using this command before it can be played again.

**[?]** This command will display the Voicing (register and voice transposition) in effect at the current cursor position. Position the cursor anywhere in the music file. Press **[BREAK]** to return to the Command Mode. At the Command Mode, type **[?] [ENTER]** and the system will display the register and voice transposition variables on the line below.

**Note:** This command, like the **@**, does a partial compilation making it necessary to re-Score the piece before it can be played again.

## Special Notes about Music Programming:

- Always make sure you are in the correct mode before you enter commands. Music Programming Mode is at the left center of the screen. The Command Mode is at the upper left corner of the screen.

- The cursor always remains on the same screen line and the file moves up and down around it.
- The insert character function **[SHIFT] [→]** adds spaces to the current line and will shift characters off the end of the line. If characters are shifted to the right of the end of the line, they are lost.
- A **[SHIFT] [↑]** will undo only one previous **[SHIFT] [↓]** character.
- Using **[SHIFT] [BREAK]** to enter the Programming Line does not clear the Command Line. This allows you to make changes, exit the Programming Mode, and search for the next occurrence of the same string without retyping the command line. Exit the Programming Mode by pressing **[BREAK]**.
- If you make a mistake while editing and have not moved the cursor from the line, **[SHIFT] [BREAK]** will cancel all changes, restore the line, and return the cursor to the beginning of the line.



## Orchestra-90 Note Conversion Chart

Tables 1 and 2 will probably be the most often used charts from the manual, especially when you begin your composing. It might be helpful if you make a photocopy of these charts and tape them near your computer for a convenient reference. Note the position of Middle C in both charts. The circled notes in Table 2 represent the Orchestra-90 Music Language Scale. The smaller letter above each circled note represents the actual note name in standard music notation.

NOTE SYMBOL	MUSICAL NOTE NAME	STAFF POSITION (see figure 3)
+G	E	
+F	D	
+E	C	
+D	B	
+C	A	
+B	G	
+A	F	
+9	E	
+8	D	
+7	C	
+6	B	
+5	A	
+4	G	
+3	F	
+2	E	
+1	D	
Ø	C	Middle-C
-1	B	
-2	A	
-3	G	
-4	F	
-5	E	
-6	D	
-7	C	
-8	B	
-9	A	
-A	G	
-B	F	
-C	E	
-D	D	
-E	C	
-F	B	
\$	REST	(any)

Table 1.

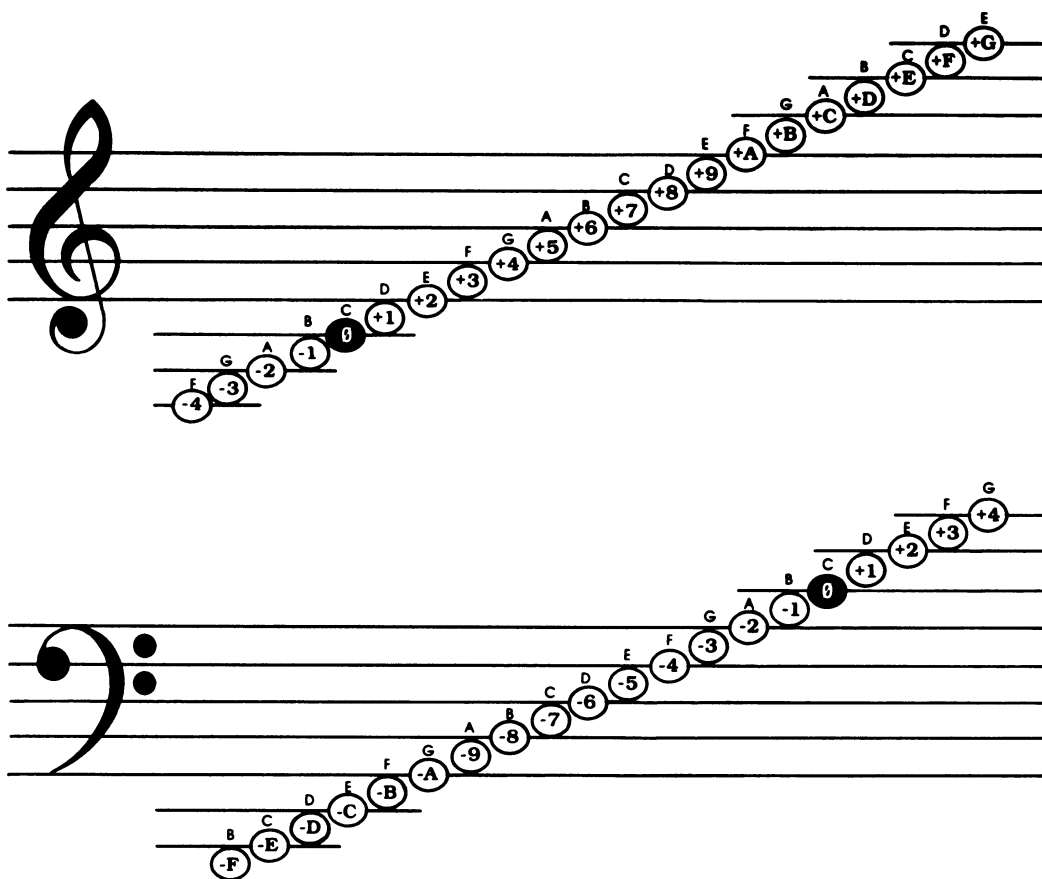


Table 2. Orchestra-90 Music Language Scale.

**Note:** The symbols inside the notes represent the **Orchestra-90** scale. The small letters above the notes represent the musical scale and are for reference only.

## Key Functions and Commands

During music programming sessions, use the following keys or key combinations to generate the specified results.

<b>BREAK</b>	Terminates the current process and enters Command Mode.
<b>CLEAR</b>	Erases the character at the cursor and everything to the right of it on the line. If you press <b>CLEAR</b> and accidentally delete a line, type <b>SHIFT</b> <b>BREAK</b> to reinstate that line.
<b>SHIFT</b> <b>CLEAR</b>	Deletes the entire line and cannot be restored.
<b>ENTER</b>	Accepts the current line as command or edit input.
<b>←</b>	Moves cursor to the left.
<b>→</b>	Moves cursor to the right.

Note: Left and right cursor movement will wrap around at the logical end of line.

<b>SHIFT</b> <b>←</b>	Deletes the character under the cursor and shifts the remainder of the line to the left. The cursor does not move.
<b>SHIFT</b> <b>→</b>	A space is inserted under the cursor by shifting the character at the cursor and the remainder of the line to the right. The cursor does not move. Any character shifted past the end of the line is lost.











While entering or editing music files at the Music Programming Line, the following control keys are used:

<b>↑</b>	Moves the cursor to the line above the current line toward the top or beginning of the file.
<b>↓</b>	Moves the cursor to the line below the current line toward the bottom or end of the file.
<b>SHIFT</b> <b>↑</b>	Joins the current line to the one above it making one long line. Position the cursor at the beginning of a line you wish joined with the one above it. Type <b>SHIFT</b> <b>↑</b> . This command is effective only if the cursor is at the beginning of a line.
<b>SHIFT</b> <b>↓</b>	Insert a new line between lines or split an existing line in half if the command is used with the cursor at the beginning of a line. If the command is used with the cursor on any other part of the line, the character under the cursor and all characters to the right will be moved down to form a new line. Type <b>SHIFT</b> <b>↓</b> .
<b>SHIFT</b> <b>?</b>	Makes a copy of the current line and inserts it at the end of the file. Position the cursor at the beginning of the line you wish to duplicate. Type <b>SHIFT</b> <b>?</b> and the line will be added to the bottom of the music file.
<b>SHIFT</b> <b>!</b>	Moves a specified line up one position toward the top or beginning of a music file. Position the cursor at the beginning of the line you wish to move. Type <b>SHIFT</b> <b>!</b> and the line will move up.

## Music Language Programming Functions

This section gives typical musical notation and the codes that the Music Language offers for each. Some of the symbols are exclusive to the system and have no traditional musical notations.

### Note Values

	<b>Whole Note</b>	Held for a duration of 4 quarter note beats in 4/4 time. The programming symbol is <b>W</b> .
	<b>Half Note</b>	Held for half the value of the whole note. The programming symbol is <b>H</b> .
	<b>Quarter Note</b>	Held for one quarter of the value of the whole note. The programming symbol is <b>Q</b> .
	<b>Eighth Note</b>	Held for one-eighth the value of the whole note. The programming symbol is <b>I</b> .
	<b>Sixteenth Note</b>	Held for one-sixteenth the value of the whole note. The programming symbol is <b>S</b> .
	<b>Thirty-second Note</b>	Held for one thirty-second the value of the whole note. The programming symbol is <b>T</b> .
	<b>Sixty-fourth Note</b>	Held for one sixty-fourth the value of the whole note. The programming symbol is <b>X</b> .
	<b>Dotted Note</b>	A period placed after the note will increase value of the note by one-half. For example, H.0 is a dotted half note played as middle C.
	<b>Triplet</b>	A colon placed in front of a note makes the note a triplet. For example, Q: says any note following will be a quarter note triplet.
	<b>Rests</b>	Rests are indicated by following a note value with a dollar sign. For example, Q\$000 indicates that there will be a rest for one beat and then three middle C quarter notes will play.

## Articulations

Orchestra-90 normally plays successive notes Legato or “tied” (smooth, with no spaces between the notes) unless Articulation is used to provide separation, distinction, and accent of notes. Articulation added to a note reduces its duration by a specific amount and replaces that amount with an equivalent rest, thus creating a space between that note and the following note.

Orchestra-90 offers four levels of articulation from an apostrophe (‘) which creates the smallest amount of separation, to a comma (,) which provides the largest amount of separation.

<b>‘</b>	<b>Short Portato</b>	An apostrophe placed after a note shortens it very slightly and creates the smallest possible separation between notes. This creates a slight accent on each note.
<b>“</b>	<b>Long Portato</b>	A quotation mark placed after a note shortens the note by an amount slightly more than Short Portato. The notes are accented more than Short Portato.
<b>;</b>	<b>Staccato</b>	A semicolon placed after a note creates the Staccato effect by shortening the note much more than Long portato. The notes have significant accent and distinction.

, **Intense Staccato** A comma placed after a note creates the Intense Staccato effect, substituting half of the note's duration with a rest to produce the strongest articulation available. The notes have the most accent and distinction possible.

## Flats, Sharps, Accidentals

#	<b>Sharp</b>	A # placed after a note signifies an accidental sharp.
b	<b>Flat</b>	An & placed after a note signifies an accidental flat.
q	<b>Natural</b>	A % placed after a note signifies an accidental natural sign.
##	<b>Double Sharp</b>	## placed after a note signifies a double sharp.
bb	<b>Double Flat</b>	&& placed after a note signifies a double flat.
q#	<b>Natural Sharp</b>	%# placed after a note signifies a natural sharp.
qb	<b>Natural Flat</b>	%& placed after a note signifies a natural flat.

( )	<b>number</b>	A programming command enclosed within parentheses followed by a single HEX digit (0-F) indicates that the enclosed sequence is to be repeated. For example, (16;SD6'')3, says to repeat the enclosed musical notation three additional times.
*		An asterisk placed in a music file line indicates that all the notes following are assumed to be played in the Treble Clef.
-		Place a - (dash) in front of a note to play only that note in the Bass Clef. This will not affect a previous Treble Clef (*) command.
/		Placing a slash at the beginning of a line lets you insert comments (e.g. a title) that will not be considered when the piece is SCORED.
<#		The < followed by a single HEX digit indicates that all the notes in the current and following measures are transposed down the number of semi-tones specified by the number.
@		The at sign indicates that all notes following are to be played in the Bass Clef.
+		Place a + (plus sign) in front of a note to play only that note in the treble clef. This will not affect a previous Bass (@) command.
—		Inserting the — ([SHIFT] [@]) indicates that the notes following are to be played in Percussion Clef.
J		J begins the Register definition. J followed by the wave form type (Random or Sinesoidal) and nine digits specifies the wave-form type, description, and volume. (See Special Instrumentation for a more complete discussion.)
K		K followed by a digit from 0-7 and # or & specifies the Key Signature, and the number of sharps or flats in the key signature. For example, K1& indicates that the key signature contains 1 flat or is in the key of F.
M		M followed by a number indicates the Measure number. For example, M12 would begin Measure 12. The compiler actually just looks for the letter M followed by any number of letters or numbers and finally a single space. Measure numbers are ignored by the system. The actual purpose of the measure string is just to tell the compiler where each new measure starts. The following are examples of valid measure strings:

M01  
 MEASURE  
 M400001X  
 M  
 MUSIC

Measure numbers are normally helpful for you when referring from a transcribed piece back to the original sheet music. For example, you could pencil-in measure numbers like M01, M02, etc. above each measure in a piece of sheet music. Then enter the numbers into the music file. If you are dissatisfied with the Played piece, or want to rearrange the piece later, use the measure numbers as a quick reference.

**Note:** When you PACK a file (see ORCHUTIL) all of the measure strings in the music file are reduced to a single M followed by a space.

**N** N followed by the note symbols (H, Q, I, S, or T) defines the time signature. For example, NQ indicates that a Quarter note gets one beat. (It is feasible to simply begin with the abbreviation, NQ = 60, then adjust the tempo of the song after transcribing and playing the first one or two measures.)

**O** Unless otherwise specified by using the O code, an accidental will affect all Voices in a measure. By entering an O1 anywhere in a measure, the accidental will affect only the voice in which it occurs.

The O2 code will carry any accidentals forward into the next measure. This is necessary when a particularly long measure must be split into two or more measures to meet the 32 note per voice per measure limit of the compiler. O2 must be set in each and every measure in which accidentals are to be carried forward.

Option O, specified O0 will cancel both O1 and O2.

**Pnumber** P followed by a 2-digit value defines the beginning of a Part numbered as the 2-digit modifier. For example, P20 indicates that a new part is begun and that the previous measure is not included in P20.

**R** R followed by a Part number indicates that the program is to repeat the NOTES in that part using the current tempo and register assignments, then proceed to the next Part. For example,

R20 V1YA V2YA V3YA V4YA NQ=80  
 R20 V1YB V2YB V3YB V4YB NQ=70

indicates that the compiler will get the notes from Part 20 and repeat them twice using new tempo and register assignments each time, then play subsequent Parts. This process of getting only the actual notes explains why the tempo and register settings may be changed for each Repeat.

If any measures follow a Repeat, they must start with a new Part number.

**U** U followed by a plus (+) or minus (-) and a single HEX digit (0-F) indicates that all the following notes in that particular voice are to be transposed up or down the number of whole steps indicated by the numbers.

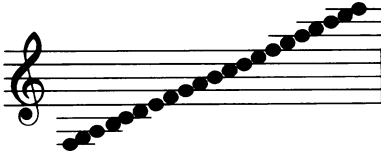
For example, V4U-7 would lower Voice 4 seven steps, or one octave.

**V** V followed by 1, 2, 3, 4, or 5 indicates which Voice you are programming. For example, V3 indicates Voice 3, which is differentiated from V2, V4, etc.

- Y** Y followed by A, B, C, D, or E follows a Voice specification and indicates that the current voice is to be a specific instrument. For example, V2YE indicates that Voice 2 is to sound like a violin. (See Special Instrumentation for a more thorough discussion of this characteristic.)
- Znumber** Z followed by the indicated one-digit number indicates that a certain Voice is to be played either from the left or right stereo channel (See Stereo Mapping.)







## 5/ Transcribing Sheet Music

The Music Language provides the means for transcribing standard musical notation into a symbolic form usable by the computer.

The use of the language can be shown best by example. Figure 5 an excerpt from J.S. Bach's *Capriccio*. Figure 6 is an excerpt from the sheet music of J.S. Bach's *Capriccio*. While the two figures may appear to have little in common, both contain essentially the same information.

```

010 /CAPRICCIO
020 /SOPRA LA LONTANANZA DEL SUO FRATELLO DILETTISSIMO
030 /J.S.BACH
040 P50/ ARIA DI POSTIGLIONE
050 K2&
060 NQ=E0 /POCO ALLEGRO
062 >2
064 V1YC V2YB V3YB
070 M1 *S6789IABSABA9I.8S7
080 V2@I$4Q1I12Q1
090 V3@Q.8I5Q48
100 M2 *(16;SD6")3
110 M3 *S7654I365S43"I3;SA3"
120 V2@Q.4"I445%Q4
130 V3@I9768Q7B
140 M4 *I3;SA3"I3;SA9&8767I.5S6"
150 V2@Q$I$4"43Q4"
160 V3@Q$I$785&Q4
170 M5 *(16;SD6")2Q6;
180 V2@Q4$$4
190 V3@Q6$$6
200 R50
210 P52
220 M6 *I8BA#BC8QB
230 V2*Q6I54S3#
240 V3@S3210*I12S1210@I.1S2
250 M7 V3@(13;S+43")2I3;+4
260 M8 *17#S89%IA987#Q8
270 V2*Q5I5654Q3
280 V3@S2345%I63456;S+16"
290 M9 *Q$I$SA3"Q3I$SA3"
300 V2@16;S+16"Q6I$SD6"QD
310 M10 *Q3I$SD6"Q6
320 V2@I$S18"Q8I$S18"I8;S18"
330 M11 *I$SD6"6789IAS67I.5S6"
340 V2@Q8I$213Q4
350 V3@Q$I$7654B
360 M12 *I6;SD6"Q6I$SD6"Q6"
370 V2@Q4I$S18"H8
380 V3@Q6$I$4Q1
390 R52

```

Figure 5.

### Nº 3. Capriccio

sopra la lontananza del suo fratello diletto.

1st eine Schmeichelei der Freunde, um denselben von seiner Reise abzuhalten.

Aria di Postiglione.

**Poco allegro.** (♩ = 76.)

The sheet music is for a piano and voice piece. It begins with a piano introduction marked 'Poco allegro' (♩ = 76). The first system shows the piano part with a triplet in the bass and a melodic line in the treble. The second system continues the piano part with various dynamics and fingerings. The third system shows the vocal entry with 'dim.' and 'cresc.' markings, followed by a piano accompaniment. The piece ends with a final cadence in the piano part.

Figure 6. Sheet Music

In Figure 5, each line begins with a three-digit number followed by a space. These line numbers are added for the purpose of this discussion only and have no bearing on the music. Remember, the contents of each line effectively begin after the first space.

Lines 10, 20, and 30 are informational only. A slash (/) appearing anywhere in a line causes the remainder of that line to be ignored.

For the purposes of transcription, a piece of music is considered to be subdivided into parts, measures, and voices. A part defines one or more measures that are played in the same key, tempo, and with the same registration. Logically, a Part usually corresponds to a portion of the piece that may be repeated, i.e., a phrase, stanza, chorus, etc. Parts are defined by the letter P followed by either a two-digit number or a space. Numbered Parts can be repeated; unnumbered parts cannot.

Measures are indicated by a character string beginning with the letter M and ending with a space. The characters following the M, usually a measure number, are ignored by the compiler and serve only as a reference between the printed musical score and the transcribed Music Language text.

A voice is a separate strand of music, in harmony or counterpoint. A trio has three voices and a quartet four, Up to five voices may be defined and they are identified by:

V1-(First Voice)  
V2-(Second Voice)  
V3-(Third Voice)  
V4-(Fourth Voice)  
V5-(Fifth Voice) For 4 MHz systems only

As a convenience, each measure begins with an implied Voice (V1) so you never have to identify V1 in the code. You can however specify a different Voice number.

In the example, Aria consists of two repeated sections, one 5 measures long and the other 7 measures long. The first section is defined in line 40 as P50. The choice of the two-digit number is arbitrary and serves only to identify the part. However, each part defined must have a unique number. Line 200 defines a Repeat of Part 50. Likewise, line 390 is a Repeat of Part 52 (line 210).

Line 50 defines the key signature. You need only specify the number of sharps (#) or flats (&). If the key signature is not specified, C major (no sharps or flats) is assumed.

Line 60 defines the time signature and tempo. NQ indicates that each quarter note gets a beat (H=half, Q=quarter, I=eighth, S=sixteenth). =C0 indicates the relative length of a beat. Normally, the tempo parameter is determined experimentally by holding various combinations of the 1-7 keys while the piece is playing. The key pattern is then translated, using the Tempo Conversion Tables, to a two digit number which is entered into the source file. It may also be necessary to alter the time signature parameter to achieve the desired tempo.

Line 62 defines transposition. The < or> character followed by a number defines the direction and number of half-steps the piece is to be transposed (<= down, > = up). While Orchestra 90 should not pose any fingering problems, it is often desirable to transpose a piece down a few semi-tones to avoid the distortion present in higher notes. This is especially true with 2 MHz systems.

Line 64 defines the tone color registers to be used by the different voices. The default register is D (organ).

Voice 1 uses register C (clarinet)  
Voices 2 and 3 use registers B (oboe)

The music proper begins in line 70 with the definition of Measure 1.

Each of the symbols representing a note in standard musical notation imply two pieces of information:

1. Its shape, along with the time signature, defines how long, relative to a beat, the note is to be held,
2. Its position on the staff, key signature, and clef, define the note to be played.

Transcribing this two-dimensional form into a single line requires two characters to represent each note.

The time value, or shape, is defined by a single letter:

W = whole  
H = half  
Q = quarter  
I = eighth  
S = sixteenth  
T = thirty-second  
X = sixty-fourth

The time value may be modified by additional symbols. Dotted notes are indicated by a period (.) following the letter. Triplet time values are indicated by a colon (:) following the letter. For example:

Q means all the notes following are quarter notes  
I. means all the notes following are dotted eighth notes  
S: means all the notes following are sixteenth note triplets

Note length modifiers may be combined:

H:.. means all the notes following are double-dotted half note triplets. (Dot modifiers cannot precede a triplet modifier. i.e., Q.: is not valid.)

The staff position of a note is defined by its relationship to a fixed point on the staff. Middle-C is always location 0 and notes above it are defined by a positive (+) displacement while notes below it are defined by a negative (-) displacement. See Tables 1 and 2. The displacement uses a hexadecimal-like number scale with the letters A to G representing the numbers 10 to 16. The maximum positive displacement is +G. The maximum negative displacement is -F. A dollar sign (\$) defines a rest.

Coding note displacements can be simplified by specifying a default displacement sign or clef. An asterisk (\*) sets the default to + or treble clef, and all unsigned notes following are assumed to be +. An at-sign (@) sets the default to or bass clef and all unsigned notes following are assumed to be -.

Accidentals are indicated with:

A sharp (#)  
A flat (&)  
Or, a natural (%) sign

Accidentals immediately follow the note affected in the Orchestra-90 music code. Accidentals stay in effect until the end of the measure or until modified by another accidental. Double sharps (##) and flats (&&) and natural sharps (%# or %#) and flats (&% or %&) are allowed. Double naturals (%%) and flatted sharps (&#) or sharped flats (#&) will give unpredictable results and should be avoided.

You should now be able to decipher all the symbols in lines 70, 80, and 90 and relate them to the first measure in Figure 6.

Line 100 introduces several new symbols. The parentheses indicate that the enclosed notes are to be repeated. The number following the right parenthesis is the number of times the information enclosed in the parentheses is to be repeated.

Line 100, although coded in the sequence, M2 \*(16;SD6")3, will play as if it was coded in the following sequence:

M2 \*I6;SD6"I6;SD6"I6;SD6"I6,SD6"

This type of repeat, known as Reiteration, should not be confused with a repeat of a Part.

Repeats, such as the type in Line 100, can be used any number of times, but cannot be nested.

For example, (((Q3;3;)))3 is unacceptable.

Articulation of notes may be changed by "expression" modifiers. These modifiers change the way a note is played without altering its pitch or overall duration. Articulation is used to introduce a small rest after a note to separate it from the note following. There are two major types of expression — staccato and portato, each type has two forms.

**Intense Staccato**, indicated by a comma (,), shortens the note by 1/2 and adds a rest equal to 1/2 the note's duration.

**Staccato**, indicated by a semi-colon (;), is similar except the note is shortened by 1/4 of its value and adds a 1/4 rest.

**Short Portato** is indicated by an apostrophe ('). This shortens the note by an amount equal to 1/3 of a 1/128 note, and adds a rest of that value.

**Long Portato**, indicated by a quote ("), is similar except the amount shortened is double, 2/3 of a 1/128 note.

In all cases, expression modifiers affect only the note they follow. Accidentals must precede expression modifiers.

Thus far, the assumption has been that the music being transcribed is in bass/treble clef notation. To take advantage of music arranged for different instruments in different clefs (soprano, alto, tenor, etc.), each voice can be defined as belonging to a different clef. Voice/clef definition is indicated by a U followed by a number that is the displacement from Middle-C in the clef being defined to Middle C in the treble clef. For example,

V1 U-2 V2 U-6 V3 U-8 V4 U-C

defines Voices 1-4 as soprano, alto, tenor and bass, respectively. Because all clefs are defined relative to the treble clef, each voice is transcribed exactly as if it were the treble clef, and no clef symbols (\* or @) need be used.

Another useful transposition would be V4U-7 to lower Voice 4 one octave to get a low bass sound. Since the most common clef transposition is negative, the default sign is "-"; e.g., U7 is the same as U-7.

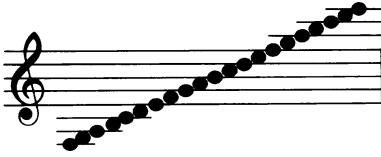
Often, when transcribing part-music, the accidentals in one voice interfere with the accidentals in another voice. This is because the compiler applies every accidental to all voices.

Option 1, specified O1, will limit the application of accidentals to the voice in which they appear.

Option 2, specified O2, will carry any accidentals forward into the next measure. This is necessary when a particularly long measure must be split into two or more measures to meet the 32 note per voice per measure limit of the compiler. O2 must be set in each and every measure in which accidentals are to be carried forward.

Option O, specified OØ, nullifies both O1 and O2.

This concludes the second transcription study. You might want to enter, play and then write the file for future reference. For practice, you could experiment with some of the Programming Commands and see how they would affect the file. Remember, when you are entering the Voicing, you can be flexible. There is no set way of Voicing a piece. Be creative and bold.



## 6/ Special Instrumentation

An orchestra could not be an orchestra if only one type of instrument played all the time. Orchestra-90 lets you create orchestral effects by giving you the flexibility to arrange your music with up to five instruments and percussion. The following information will help you enhance your final musical product by showing:

- Default instrumentation
- How to change the instrumentation
- How to use the percussion instrumentation
- Stereo mapping

### Changing Instrumentation (Tone Color Registers)

The default instrumentation is based on the spectral analysis of orchestral instruments and was chosen to provide a wide range of musical sounds. The registers are easily altered to imitate other standard instruments or to create unusual, electronic sounds. You have the flexibility to use harmonic or percussive sounds or to combine the two by utilizing creative voicing.

A register definition consists of the following series:

- The letter J
- The initial of the register name (A, B, C, D, or E)
- The initial of the wave form type (R or S)
- The parameter list consisting of eight digits representing the weight of partials #1 through #8
- A single digit representing volume

The standard registers are all harmonic or waveform type S (Sinesoidal). Percussive sounds may be obtained from waveform type R (Random) as well as type S.

The harmonic registers are defined by the sum of eight sine waves or partials. Each partial is an integer multiple of the fundamental frequency. The first partial is the fundamental, the second is two times the fundamental, the third is three times, etc. The eight digits in the parameter list define the relative strength or weight of each partial in the register.

This section provides a reference for you when selecting the appropriate instrumentation for whatever piece you are programming. It's easy and fun to experiment with each available instrument. Don't be surprised if you arrive at some unorthodox tones when experimenting with instrumentation. The following discussion illustrates the default registers for both harmonic and percussion instruments and also should provide a guide you can use to create your own sound.

The standard default registers are:

JASEFA50000E	/Define register A, <i>trumpet</i>
JBS48F80000F	/Define register B, <i>oboe</i>
JCSE0500000A	/Define register C, <i>clarinet</i>
JDSF4080000B	/Define register D, <i>organ</i>
JES4F280000D	/Define register E, <i>violin</i>

The default registers at 4 MHz clock rate are:

JASEFA54E00E     /Define register A, *trumpet*  
 JBS48F8F200F     /Define register B, *oboe*  
 JCSE0S0F000A     /Define register C, *clarinet*  
 JDSF4080000B     /Define register D, *organ*  
 JES4F281400D     /Define register E, *violin*

A trumpet-like sound, standard default Register A, is defined as follows:

JASEFA50000E	
Symbol	Meaning
J	Begin register definition
A	Define register A
S	Sinesoidal wave (harmonic)
E	Partial #1, fundamental, weight E (nearly maximum)
F	Partial #2, first harmonic, weight F (maximum)
A	Partial #3, second harmonic, weight A
5	Partial #4, third harmonic, weight 5
0	Partial #5, fourth harmonic, weight 0
0	Partial #6, fifth harmonic, weight 0
0	Partial #7, sixth harmonic, weight 0
0	Partial #8, seventh harmonic, weight 0
E	Volume E (nearly maximum)

### Default register for a trumpet.

Each register definition must contain weights for all eight partials. Partial with a weight of 0 make no contribution to the final waveform.

Because it takes a considerable amount of time to generate a harmonic wave, the compiler will do so only when the register definition changes. You will notice that the first piece compiled takes an unusually long time. This is because all five default registers are being generated. All registers are fully defined to default values unless redefined in the music file.

To redefine one of the default registers simply enter a complete register definition at the beginning of the music file:

JASEFA50000F

The only difference in this register as compared to the standard default register is that the volume has been increased to F, which is slightly louder than the default volume of E. You could also change the weighting of the partials. You can make a voice silent by changing all the partials to 0. Unless you are attempting some special effects, partials 5-8 should be weight 0 if you are using the standard Model III clock rate of 2 MHz. Special register definitions are to be saved along with the rest of the music file.

If you have a TRS-80 with 4 MHz modification and you answered “Y” to the FAST CLOCK question upon power-up or if you are using a TRS-80 Model 4, two Partial have been automatically added to the tone color registers.

## Percussion Instrumentation

Percussion and other special effects may be obtained by defining a register as R which indicates that a pseudo-random number generator is to be used. The eight digits in the parameter list define the four-digit random seed and the four digit “randomizing” function, respectively. You can experiment by changing these eight digits to create your own unique effects. Because there are billions of combinations of seeds and functions, few generalizations can be made about the kinds of percussive effects available.



Here are two samples:

JBR000300058 /Register B, scratchy sounds  
JCR101000018 /Register C, squeaky sounds

In both cases the volume is 8.

The Sinesoidal waveform can also be used for percussion:

JES80011001E

This register definition will be used for percussion examples later. Just as you would not attempt to play a melody on a drum, you should not try to play music on a percussive register.

Regardless of the waveform you use, there is a special way to play percussion, which requires a unique clef.

The Percussion Clef, defined by an underline ( \_ ), has 16 notes defined by the unsigned hex digits 0-9 and A-F. The underline character is produced with a **SHIFT** **@**. This clef also inverts the function of the articulation. The usual operation of articulation is to shorten a note by a very small amount and insert a small rest at the end.

While in the Percussion Clef, the note is made very short and a large rest is inserted at the end. This Reverse Articulation creates the short percussive sound of a drum and allows the entry of drum rhythms using the same note values found in drum tablature. The” and’ symbols are the best to use for reverse articulation in the Percussion Clef. The following score will demonstrate the features of the Percussion Clef.

Symbol	Meaning
JER80011001A	/Define register E, random waveform, volume A
JCS80011001F	/Define register C, sine wave form, volume F
	Other registers default
	Articulation on all notes
P00 <7	/Unnumbered Part, piece played one octave down
	Play standard scale using default register D
M01 V1YD *I0’1’2’3’4’5’6’7’8’9’A’B’C’D’E’F’Q\$	/Same scale using random wave form without Percussion Clef
M02 V2YE *I0’1’2’3’4’5’6’7’8’9’A’B’C’D’E’F’Q\$	/Same scale using random wave form with Percussion Clef
M03 V2YE _I0’1’2’3’4’5’6’7’8’9’A’B’C’D’E’F’Q\$	/Same scale using sinesoidal wave form and Percussion Clef
M04 V3YC _I0’1’2’3’4’5’6’7’8’9’A’B’C’D’E’F’Q\$	

#### Percussion Demonstration.

In reviewing the sample, Measure 1 (M01) sounds normal. But, M02 sounds very unusual because the notes are not shortened using reverse articulation. M03 and M04 are examples of proper use of register definition, reverse articulation, and Percussion Clef. Once you have defined a percussion register you may wish to play the entire scale using the Percussion Clef and select notes which sound best. Some notes in any percussion register may have no sound at all or may not be suited for a particular piece.

Remember that the underline character ( \_ ) defines default Percussion Clef with Reverse Articulation and all notes following will be percussion notes until the (\*) or (@) sign is encountered.

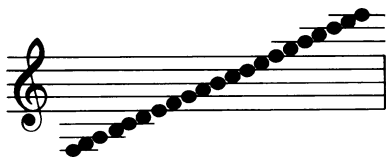
A music Voice following a percussion Voice must be defined by (\*) (Treble Clef) or (@) (Bass Clef). Typically, the last Voice in any measure is used for percussion with the first Voice in the following measure being defined as either (\*) or (@).

## Stereo Mapping

Stereo mapping allows you to balance or position voices in each stereo channel or to “pingpong” voices between channels. The mapping symbol is Z followed by a number from 0 to 9. The default is Z0.

Symbol	Channel A		Channel B		Synthesizer	
Z0	V1	V2	V3	V4	V5	3, 4, or 5
Z1	V1	V3	V2	V4	V5	3, 4, or 5
Z2	V2	V3	V1	V4	V5	3, 4, or 5
Z3	V1	V4	V2	V3	V5	3, 4, or 5
Z4	V2	V4	V1	V3	V5	4 or 5
Z5	V3	V4	V1	V2	V5	4 or 5
Z6	V1	V5	V2	V3	V4	5
Z7	V2	V5	V1	V3	V4	5
Z8	V3	V5	V1	V2	V4	5
Z9	V4	V5	V1	V2	V3	5

The simplicity of the table hides the difficulty of the implementation. The compiler actually performs the mapping in two parts. Voice mapping takes place at Measure boundaries. This means that if the mapping symbol appears somewhere other than the beginning of a Part, a voice may become disconnected from its register. To put it another way, if all voices are playing the same register, mapping may be specified at measure boundaries, otherwise the results are unpredictable.



## 7/ Music File Processing (ORCHUTIL)

This section describes the music file utility program, ORCUTIL/CMD. Music files can be transferred to disk using ORCHUTIL. (Refer to Transferring Cassette Music Files to Diskette.) ORCHUTIL can be used:

- To transfer music files between different media (Tape to disk or vice versa)
- To perform format conversion (Binary to ASCII and the reverse) which comes in handy when sending or receiving files through telephone lines using a modem.
- To make a one or two-character global change.

Before using ORCHUTIL, make sure the cassette tape version of Orchestra-90 is properly inserted into your cassette player and is rewound. To load ORCHUTIL, at **TRSDOS Ready** type:

**ORCUTL**

After a few moments the ORCHUTIL menu displays:

```

ORCHUTIL - Copyright 1981, 1982, 1983 John Bokelman
Licensed to Tandy Corporation, All rights reserved.
Memory File = <none>
Input mode = Binary format disk (/ORC)
Output mode = Binary format disk (/ORC)
  I : select Input      O : select Output
  R : Read              W : Write file
  M : Modify x to y    C : Change XX to YY
  P : Pack file        K : Kill file
  D : Directory        Q : Quit
Type the letter corresponding to desired function -> _

```

### **I : select Input / 0 : select Output**

These two options let you select the desired Input or Output mode to be used for all subsequent file transfers. The Input or Output mode is for the selection of the baud rate for cassette users or for the character conversion type (Binary or ASCII) for disk users. Upon selection, the mode in effect is displayed in the second and third status lines:

```

Input Mode = Binary format disk (/ORC)
Output Mode = Binary format disk (/ORC)

```

- Press  or  for the following Menu:

```

  L : Low speed tape (500 baud)
  H : High speed tape (1500 baud)
  B : Binary format disk (/ORC)
  A : Ascii format disk (/ASC)

```

Type the letter corresponding to desired function -> \_

The following options depend on the type of system you are using:

- Press **L** to read and write compatible Model I tapes. This option can be used with any system.
- Press **H** to read and write compatible Model III tapes. This is valid only on a Model III with high speed support enabled.

- Press **B** to read and write standard Orchestra-90 disk files. This option is only valid for a Model III or 4 disk system. The file extension is always **/ORC**.
- Press **A** to read and write Orchestra disk files in an ASCII format suitable for transfer via modem. This option is only valid for a Model III or 4 disk system. The file extension is always **/ASC**.

ORCHUTIL performs all the necessary format conversions as the file is read or written. You may read in one mode and write in another in any combination supported by the hardware.

## **R : Read / W : Write a File**

Selecting either item will provide a prompt for the name of the file to be read or written. If the current file mode is tape, the file name should be six characters or less with no spaces between characters. A file name longer than six characters will only contain the first six characters when written to tape. An asterisk (\*) in the file named will match any character(s). Reading a file named \* will read the next file on the tape.

If the current file mode is disk, the file name should be eight characters or less with no spaces, optionally followed by a colon (:) and a drive number.

After a file is read, its name will appear in the first status line as the current memory file. The Write command will not be accepted unless there is a valid file in memory.

To load a music file, press R.

At the bottom of the screen you are asked:

Enter INPUT file name > \_

- Type the name of the file you wish to process and press **ENTER**.

If the file is located correctly, the file name will appear at the line, Memory file = filename. If an error is made when entering the file name, the message, **I/O error**, appears in the Memory file line. If this error occurs, you can attempt to Read the file again by pressing **R** and re-entering the file name.

To Write a file, press W.

The prompt displays:

Enter OUTPUT file name -> \_

- Type the file name and press **ENTER**.

You can then process another file.

## **M : Modify x to y and**

## **C : Change xx to yy**

These options let you globally change any one or two-character sequence in the file. Selecting either will result in the prompts xy and xxyy, respectively. In either case, x represents the character(s) to match and y is the replacement character(s). The replacement operation considers the file as one character string; line boundaries are ignored. When a match is found, the character(s) is replaced and the scan continues with the next character in the file. These commands will not be accepted unless there is a valid file in memory and the xy or xxyy strings are entered with precisely 2 and 4 characters respectively.

Press **M** to modify one character. The prompt displays:

Enter xy (two bytes) -> \_

- Type one character presently in the file followed by the character you wish to change it to and press **ENTER**.

For example, type **AB** **[ENTER]** to change every occurrence of A to B.

or

Press **C** to modify two characters.

The prompt displays:

Enter xxyy (four bytes) -> \_

- Type two characters presently in the file followed by the two replacement characters and press **[ENTER]**.

For example, type **AABB** **[ENTER]** to change every occurrence of AA to BB

In order to permanently change the file, you will need to Write the file as explained earlier.

## **P : Pack file**

The Pack command is used to reduce the size of a music file by removing all unnecessary measure numbers and spaces. A “packed” file can be sent by modem more effectively. If there is not a valid file in memory, the Pack attempt will be rejected.

To Pack a file, make sure that you have successfully Read a valid file into memory. Press **P** and the system will automatically Pack the file.

To permanently change the file as “Packed” you will need to Write the file as explained earlier.

## **K : Kill file (for Disk Systems)**

The Kill file option is used to delete a file from the disk’s directory. Press **K** and the prompt displays:

Enter filespec -> \_

- Type the complete filename including the extension **/ORC** or **/ASC** and press **[ENTER]**. You can then process another file.

## **D : Directory**

The Directory command is available only for Disk Systems and is used to display the music files on a specified drive. Press **D** and the prompt displays:

Enter drive number -> \_

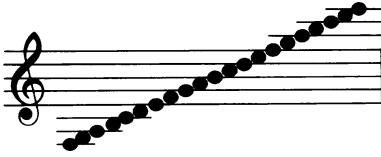
- Type the disk drive number.
- After you have satisfactorily examined the files on that disk, press **[ENTER]** to return to the Main Menu.

## **Q : Quit**

To exit ORCHUTIL and return to TRSDOS READY, press **Q**.

Note: While using one of the above options, you can press **[BREAK]** to stop the processing. Then, you can select another option.





## 8/ Technical Information

### Theory of Operation

This synthesizer uses a sampling technique commonly used in professional digital synthesizers.

The Sampling Theorem tells us that any wave form, no matter how complex, can be reconstructed from a rapid succession of discrete voltages. The reconstructed wave form will actually be a stepped approximation of the original. If the steps are small enough they will not be noticed. The size of the steps is determined by the sample rate and the frequency of the wave being reproduced.

Consider one cycle of a simple sine wave. If its amplitude was measured as 100 equally spaced intervals, the values could be recorded in a table. We could reproduce it by serially setting a voltage equal to each entry in the table. Since the table represents exactly one cycle, the frequency of the synthesized wave will be equal to the sample rate divided by 100 (the number of entries in the table). At 1000 samples per second, the entire table will be accessed 10 times per second, producing a 10 Hertz sine wave. Doubling the sample rate will double the frequency of the synthesized sine wave.

Using the same table and sample rate we could effectively double the frequency of the synthesized wave by accessing every other table entry. This gets us through the entire table twice as fast. Similarly, taking every third or fourth table entry will triple or quadruple the frequency. This is the basis of the synthesizer, except to get musically accurate frequencies, it is necessary to skip a fractional number of table entries.

Theoretically, the highest frequency that can be synthesized is equal to half the sample frequency. This is called the Nyquist frequency. As this limit is approached, the synthesized wave form becomes more and more distorted since there are fewer samples in each cycle. As the distortion increases, a secondary tone, or alias, is produced. If the Nyquist frequency is exceeded, the synthesized frequency is replaced by its alias. This phenomenon is not limited to the fundamental frequency. It affects each component of the synthesized wave form.

Aliasing cannot be filtered out, it can only be avoided. If aliasing is a problem, redefine the registers with fewer partials or transpose the entire piece down a few half-steps. An alternative solution would be to speed up your CPU.

### Compiler Organization (SCORE)

This section describes certain features of the Compiler in more detail. A little bit of technical information will help in understanding how the compiler works and how certain parameters are interpreted.

The output of the Compiler consists of two lists: a Part List and a Note List. Each element in the Part List contains a pointer to a Note List element and parameters, such as tempo and registration, that describe how the Note List is to be played. Each element in the Note List contains note frequencies and durations for each of three, four, or five voices.

During compilation, the source file is processed and the data collected is placed into either a Part Buffer or a Note Buffer. At the appropriate time, the contents of the Buffers are processed into the corresponding List.

When compilation begins, all Buffers are cleared and the default values are entered. If written out, the default values would look like this:

P00	<i>un-numbered part</i>
K0#	<i>key of C-major, no sharps or flats</i>
<0	<i>no transposition, up or down</i>
NQ	<i>time signature based on a quarter note</i>
=C0	<i>each beat has a value of C0 HEX</i>
V1 YD U0	<i>voice 1 uses register D, no clef transposition</i>
V2 YD U0	<i>voice 2 uses register D, no clef transposition</i>
V3 YD U0	<i>voice 3 uses register D, no clef transposition</i>
V4 YD U0	<i>voice 4 uses register D, no clef transposition</i>
V5 YD U0	<i>voice 5 uses register D, no clef transposition</i>
JASEFA50000E	<i>define register A, trumpet</i>
JBS48F80000F	<i>define register B, oboe</i>
JCSE0500000A	<i>define register C, clarinet</i>
JDSF4080000B	<i>define register D, organ</i>
JES4F280000D	<i>define register E, violin</i>
MV1 *	<i>begin a measure with voice 1 in treble clef</i>

The contents of the Part Buffer are added to the Part List whenever a PART or REPEAT symbol is processed or the end of the file is reached. The contents of the Note Buffer are added to the Note List whenever a MEASURE, PART, or REPEAT symbol is processed or the end of the file is reached.

Referring to Figure 5, PART symbol in line 40 will cause the contents of the Part Buffer to be added to the initially empty Part list and the Note Buffer to be added to the initially empty Note List. A new Note List element is started and the Note list pointer in the Part Buffer is changed to point to it.

Lines 50 to 64 modify most of the default values in the Part Buffer. In line 70, the MEASURE symbol will cause the Note Buffer to be processed and the Note frequencies to be transferred to the Note List. The Note Buffer is cleared and any accidentals are reset. (Since the Note Buffer is empty nothing of interest has happened yet.) The notes defined in lines 70, 80, and 90 are then placed in the Note Buffer.

The MEASURE symbol in line 100 causes the Note Buffer to be processed using the key signature and transposition parameters in the Part Buffer, and the calculated note frequencies of Measure 1 are transferred to the Note List. The Note Buffer is cleared, accidentals reset and the rest of the line is transferred to the Note Buffer.

This process of accumulating notes in the Note Buffer and processing them only when a new measure is defined continues until line 200. The REPEAT symbol causes the Note Buffer to be processed and Measure 5 is added to the Note List. It also causes the Part Buffer to be added to the Part List and a new Note List element to be started. The Note List pointer in the Part Buffer is changed to point to the same Note List element that PART 50 pointed to.

The PART symbol in line 210 acts much the same as that in line 50. Similarly, line 390 is like line 210 and the end of the file forces out the remaining buffer contents.

The result of the compilation is a Note List with the following elements:

<1>	(empty)
<2>	measures 1, 2, 3, 4, 5
<3>	(empty)
<4>	measures 6, 7, 8, 9, 10, 11, 12
<5>	(empty)



and a Part List with elements and Note List pointers as follows:

```
<00> points to <1>
<50> points to <2>
<00> points to <2>
<52> points to <4>
<00> points to <4>
```

Notice that Note List elements 3 and 5 are not pointed to. These are the elements created when the REPEAT symbol is processed. Any notes defined between a REPEAT and a PART symbol are processed into one of these “unclaimed” elements and are never played. In all other respects, REPEATS are like unnumbered PARTs.

REPEATS can alter any of the parameters in the Part Buffer and thereby repeat sections of music with a different tempo and/or registration. For example, the following will play a sample scale using each of the different tone color registers.

```
P01  YA      S012343210$ /register A
R01  YB                      /register B
R01  YC                      /register C
R01  YD                      /register D
R01  YE                      /register E
```

The default values NQ=C0 and V1\* are assumed.

## Parameters

Parameters take effect when data is processed from the buffer to the list. All parameters remain in effect until changed, explicitly or by default.

Parameters affecting the Key Signature (K and O1) or Transposition (<, >, and U) take effect at the beginning of the current Measure and stay in effect until changed.

Accidentals take effect immediately and stay in effect until changed or the end of the current measure unless Option 2 (O2) appeared within the measure. (Accidentals are also reset by K.)

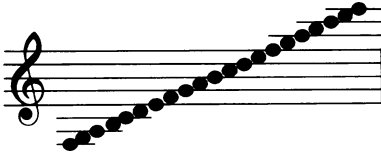
Parameters affecting the tempo (N and =) or Registration (Y) take effect at the beginning of the current Part and remain in effect until changed.

Parameters affecting the Clef (\*, @, and \_) or Note duration (W, H, Q, I, S, T, or X, with or without modifiers, . or :) take effect immediately and remain in effect until changed or the beginning of the next Measure. Measures always begin with a default V1.

Register definition (J) takes effect at the beginning of the piece. Each register may be redefined only once per music file.

The stereo mapping parameter (Z) has two different actions that take effect at different times. Voice mapping takes effect at the beginning of the current measure and remains in effect until changed. Register mapping takes effect at the beginning of the current part and remains in effect until changed.





## 9/ Converting from Cassette to Disk

The cassette tape that comes with the package includes the program that is intended to be used with disk equipped systems. On one side of the tape is the **ORC90T** program which is to be used with cassette recorder.

On the other side of the tape is the **ORCH90** program which must be transferred from tape to disk. Also, the **ORCHUTIL** program must be transferred from tape to disk.

To transfer the **ORCH90** program and the **ORCHUTIL** program from tape to disk follow these steps:

- 1 Power-up the system and make sure that the cassette recorder is correctly connected to the computer. Insert the Orchestra-90 cassette into the cassette recorder making sure that the ORCH90 label is up. Rewind the tape. Set the cassette recorder to PLAY.
- 2 Insert a TRSDOS 1.3 disk into Drive 0 and press the **RESET** button. (Model 4 owners should also use a TRSDOS 1.3 disk, not 6.0.)
- 3 When TRSDOS READY displays, type:

**TAPE (S=T,D=D) ENTER**

- 4 The prompt displays:

CASS?

**Type H ENTER**

The screen shows:

Device = Tape to Disk

Press any key when Cassette ready

- 5 Press any key other than **BREAK**. The tape will then load. The program name will display on the screen as the tape begins loading. The program will be saved to disk automatically.

Repeat this operation for both **ORCH90** and **ORCUTL**. To run either program, return to TRSDOS Ready and enter the program name.

After you transfer the **ORCUTL** program to diskette, you can transfer the music files on cassette to diskette. Refer to Transferring Music Files on Cassette to Diskette for instructions.

## Transferring Music Files on Cassette To Diskette (Disk Systems Only)

The music files on the Orchestra-90 cassette can be transferred to diskette using the **ORCHUTIL** program. If you plan to do this you must transfer the **ORCHUTIL** program to diskette prior to transferring music files. The music file-to-disk instructions can be separated into three steps. First, you set the input-output settings for the file transfer. Next, you will **READ** a music file from the cassette, which stores the file in the computer's memory. Then, you **WRITE** the file to diskette. To transfer a music file:

1. Insert your system diskette with the **ORCHUTIL** program into Drive 0. Type:

**ORCUTL** **[ENTER]**

2. The ORCHUTIL menu will display. First, the Input-Output instructions need to be set.

- Type **I** **[ENTER]** for Input mode.
- Type **H** **[ENTER]** for High Speed.
- Type **O** **[ENTER]** for Output mode.
- Type **B** **[ENTER]** for Binary format.

3. Now a file must be loaded into the memory using the Read option. Make sure the cassette recorder is properly connected to the computer. Insert the Orchestra-90 cassette into the tape recorder so that the label with the music file names is up. Rewind the tape. The cassette recorder should be set to **PLAY**.

- Press **R**. The prompt displays:

Enter INPUT file name -> \_

4. This prompt is asking you to enter the file name you wish to Read from the cassette. The first music file on the cassette is **GYPSY**.

Type **GYPSY** **[ENTER]**

The cassette recorder will then start. When the cassette recorder stops, you will know that the music file has been loaded into memory.

5. Now you can **WRITE** the music file to diskette.

- Press **W**. The prompt displays:

Enter OUTPUT file name -> \_

6. Type **GYPSY** **[ENTER]**. The disk drive will start. When the disk drive stops, you will know that the file has been saved.

7. At this point, to load another music file you do not have to enter the Input-Output settings nor Rewind the cassette tape.

The next music file on the tape is **ENTRY**.

- Press **R** to Read the file. The prompt displays:

Enter INPUT file name -> \_

- Type **ENTRY** **[ENTER]** and after a few seconds the file will be loaded into memory.

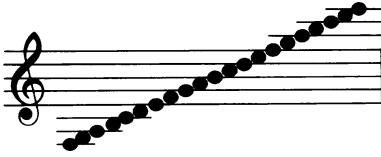
8. To save the file to diskette, press **W**. The prompt appears:

Enter OUTPUT file name -> \_

- Type **ENTRY** **ENTER**. The file will be saved to disk.

9. Repeat Steps 7 and 8 for the two remaining files on the cassette which are **CZAR** and **RACES**.





## 10/ Configuring the Software for a Model III 4 MHz System (not for Model 4 users)

The modifications codes included in this section are samples. At the time your system was altered for 4 MHz operation you should have received the codes for altering the clock speed. The following sequence begins at the program loading stage:

1. Power-up the system. Insert and rewind the cassette containing the programs for the tape version of Orchestra-90.

2. The first prompt displays:

CASS?

- Type:

**H** **[ENTER]**

3. The prompt displays:

Memory Size?

- Type:

**[ENTER]**

4. The prompt displays:

Radio Shack Model III basic  
READY  
>

- Type:

**SYSTEM** **[ENTER]**

5. The prompt displays:

\*?

- Type:

**ORC90T** **[ENTER]**

6. The prompt displays:

\*?

- Type:

**/** **[ENTER]**

When the program starts, the screen will be cleared, a copyright notice will appear on the bottom line and the first question in the configuration sequence will appear in the middle of the screen with a blinking cursor on a line below it.

7. The prompt displays:

DUPLICATE?<Y/N>

You can either:

- Press N if you choose not to make a backup of the program, or

- Make sure that you have a blank cassette in the cassette recorder and that the recorder is in the RECORD mode. Press Y(enter) After a few seconds, the program is copied. The blinking cursor returns.

8. The prompt displays:

FAST CLOCK?<Y/N>

- As your TRS-80 has been modified for 4 MHz operation, press:

Y **ENTER**

9. The prompt displays:

SOFTWARE CONTROL?<Y/N>

There are two ways to activate the clock speed-up modification — via hardware or software. Hardware clock speed-up modification can usually be initiated using a switch. Software modification is controlled by software. In response to this prompt:

- Type Y **ENTER** if your speed-up modification is controlled by software.
- Type N **ENTER** your speed-up modification is controlled by hardware, such as a switch. If so, proceed to step 11.

10. The prompt displays:

ENABLE CODE?

To make optimum use of a software controlled clock modification the program needs to know the instructions to execute to enable the fast clock. Enter those instructions, up to 8 bytes and in Hexadecimal. (Orchestra-90 will only accept the HEX Machine Code.)

Some example modifications would be:

OUT (254),1 **ENTER** (in BASIC)

or

LD A,1 **ENTER**

OUT (254),A **ENTER** (in assembly language)

or

3E01D3FE **ENTER** (in HEX Machine Code)

11. The prompt displays:

DISABLE CODE?<Y/N>

It is usually necessary to disable the fast clock before reading or writing to tape or disk. If you tell it how, the program will switch to the standard clock rate before doing any I/O. Enter those instructions, up to 8 bytes in Hexadecimal. (Orchestra-90 will only accept the HEX Machine Code example.)

Several standard modification codes are:

OUT254,0 **ENTER** (in BASIC)

or

LD A,0 **ENTER**

OUT (254),A **ENTER**

or

3E00D3FE **ENTER**

If you wish to do I/O with a fast clock, enter an NOP instruction: 00

12. The prompt displays:

HOW MANY VOICES?<3/4/5>



- With your system altered for 4 MHz, type either **3**, **4**, or **5** and then **[ENTER]**. You are not required to select five voices. The fewer voices selected, the less aliasing, the higher the resolution, and the less memory required per music file.

13. The prompt displays:

SAVE PROGRAM?<Y/N>

The configured program is now in memory and is ready for use. You can:

- Type Y **[ENTER]** if you want to permanently set the configuration to the previous entries. If you press Y **[ENTER]**, an additional prompt will appear asking you to enter a PROGRAM NAME. Enter a file name to be used to save the program. For example, a three voice synthesizer based on a fast clock might be named ORCH3F

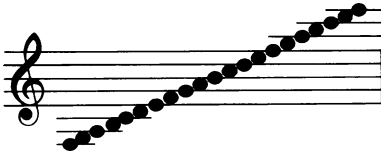
**Note:** **For Cassette Users:** the maximum length of a program name is six characters.

**For Disk Users:** the **maximum** length of a program name is eight characters followed by a drive number. The /CMD extension is automatically appended to the program name.

Press N **[ENTER]** for the SAVE PROGRAM question and end the configuration session. The blinking line cursor will be at the upper left corner of the screen.

Return to the section of this manual, A Mini-Concert, to learn the fine points of arranging and composing music,





## 11/ Error Messages

When an error is detected, an error number will be displayed. Whenever possible, the cursor will be positioned at, or near the place where the error was encountered.

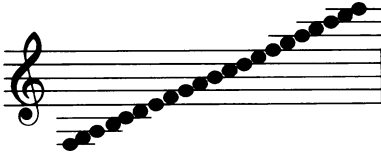
Error Message	Meaning
<b>Err 1: MEMORY OVERFLOW</b>	<p>There is not enough memory to SCORE the current file, even after OVERLAPPING.</p> <p>There is not enough memory to EDIT the current file. Consider expanding your system or splitting the file into two or more smaller sections.</p> <p>There is not enough memory to READ or GET the requested file. Expand the memory in your system.</p>
<b>Err 2: SYMBOL OUT OF</b>	<p>The command operand contains an invalid hexadecimal digit.</p> <p>The TEMPO symbol = is not followed by two hexadecimal digits.</p> <p>The compiler was expecting a hexadecimal digit or a note symbol (0-9 or A-G) but none was found.</p>
<b>Err 3: PARAMETER ERROR</b>	<p>The number of sharps or flats in the KEY signature is too large or is not followed by SHARP (#) or FLAT (&amp;) symbol.</p> <p>The note value in the TIME signature is not H, Q, I, S, or T.</p> <p>The VOICE specified is not 1, 2, 3, 4, or 5.</p> <p>The REGISTER specified is not A, B, C, D, or E.</p>
<b>Err 4: INVALID PART NUMBER</b>	<p>The PART or REPEAT specified is not a two digit hexadecimal number or the PART is not blank. P00 is considered an unnumbered part.</p> <p>The PART is already defined.</p> <p>The REPEAT is not previously defined.</p> <p>The piece cannot be PLAYED because the PART cannot be found or because the piece needs to be SCORED.</p>
<b>Err 5: MEASURE OVERFLOW</b>	<p>The measure being compiled contains more than 32 notes per MEASURE voice. Split the measure in half and marry forward any accidentals.</p> <p><b>Note:</b> Staccato and portato articulations generate a note and a rest so mount them as two notes.</p>
<b>Err 6: NOTE OVERFLOW</b>	<p>Dotted whole notes, 7-dotted triplets and other unusual combinations cannot be compiled.</p>
<b>Err 7: FILE I/O ERROR</b>	<p>The host operating system reported an I/O or other error during the last file operation.</p> <p>READ or GET: File not found.</p> <p>WRITE: Disk or directory full.</p> <p>READ, WRITE, GET, or DIR: Unrecoverable I/O error.</p> <p>DIR: Invalid device number.</p>

---

**OVERLAP?**

If there is not enough memory to hold both the source file and the compiled object code at the same time, the program will pause and display the question OVERLAP? in the status line. If you answer Y, compilation will resume and portions of the source file will be OVERLAPed by the object code. Answering anything else will cancel the compilation and enter the command mode.

This function is useful when compiling large files as the end of the object code can OVERLAP the beginning of the source code. DO NOT OVERLAP a file that has not been saved on disk or tape since this function implies the probable destruction of the current file. Do not expect to be able to use it in any way once this command is given.



## 12/ Tempo Conversion Chart

By pressing the numeric keys 1-7 you can change the tempo of a piece while it is playing. Release the keys to play the piece at the tempo in the Music Language code. This is an excellent method to experiment and find tempo at which a piece sounds best. Examine the Tempo Conversion Chart. The more keys pressed down simultaneously, the slower the tempo. For example, pressing keys 1-7 makes the tempo extremely slow.

Once the correct tempo has been established by experimentation, the corresponding hexadecimal value should be transferred to the source file. The Tempo Conversion Chart illustrates each tempo change and gives the Hexadecimal value you would enter in the code for that particular tempo.

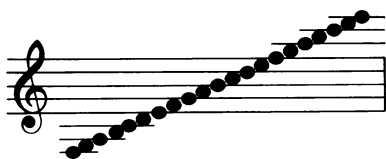
For example, if your experimentation proves that your piece sounds best when you press down keys 1,3,4,5,6, and 7, you can return to the programming line while the mode is displayed on your screen and change the tempo specification. In this case, the chart shows that those keys correspond to HEX FA. Go to the tempo specification in the Music Language mode and change it to FA. An example would be NQ = FA.

Tempo settings below 80 HEX may cause an undesirable shift in frequency and should be avoided. It may be necessary to adjust the time signature parameter to get the tempo between 80 and FE. Increasing the note value of the time signature will allow you to double the value of the tempo. For example, NQ = C0 is preferable to NI = 60.

The 0 key is used as a stop switch. When that key is depressed, the synthesizer will stop and the program will return to the Command Line.

DIGITS	HEX	DEC	DIGITS	HEX	DEC	DIGITS	HEX	DEC
1234567	FE	254	12345-7	BE	190	123456-	7E	126
-234567	FC	252	-2345-7	BC	188	-23456-	7C	124
1-34567	FA	250	1-345-7	BA	186	1-3456-	7A	122
--34567	F8	248	--345-7	B8	184	--3456-	78	120
12-4567	F6	246	12-45-7	B6	182	12-456-	76	118
-2-4567	F4	244	-2-45-7	B4	180	-2-456-	74	116
1--4567	F2	242	1--45-7	B2	178	1--456-	72	114
---4567	F0	240	---45-7	B0	176	---456-	70	112
123-567	EE	238	123-5-7	AE	174	123-56-	6E	110
-23-567	EC	236	-23-5-7	AC	172	-23-56-	6C	108
1-3-567	EA	234	1-3-5-7	AA	170	1-3-56-	6A	106
--3-567	E8	232	--3-5-7	A8	168	--3-56-	68	104
12--567	E6	230	12--5-7	A6	166	12--56-	66	102
-2--567	E4	228	-2--5-7	A4	164	-2--56-	64	100
1---567	E2	226	1---5-7	A2	162	1---56-	62	98
----567	E0	224	----5-7	A0	160	----56-	60	96
1234-67	DE	222	1234--7	9E	158	1234-6-	5E	94
-234-67	DC	220	-234--7	9C	156	-234-6-	5C	92
1-34-67	DA	218	1-34--7	9A	154	1-34-6-	5A	90
--34-67	D8	216	--34--7	98	152	--34-6-	58	88
12-4-67	D6	214	12-4--7	96	150	12-4-6-	56	86
-2-4-67	D4	212	-2-4--7	94	148	-2-4-6-	54	84
1--4-67	D2	210	1--4--7	92	146	1--4-6-	52	82
---4-67	D0	208	---4--7	90	144	---4-6-	50	80
123--67	CE	206	123---7	8E	142	123--6-	4E	78
-23--67	CC	204	-23---7	8C	140	-23--6-	4C	76
1-3--67	CA	202	1-3---7	8A	138	1-3--6-	4A	74
--3--67	C8	200	--3---7	88	136	--3--6-	48	72
12---67	C6	198	12----7	86	134	12---6-	46	70
-2---67	C4	196	-2----7	84	132	-2---6-	44	68
1----67	C2	194	1-----7	82	130	1----6-	42	66
-----67	C0	192	-----7	80	128	-----6-	40	64

**Tempo Conversion Chart.**

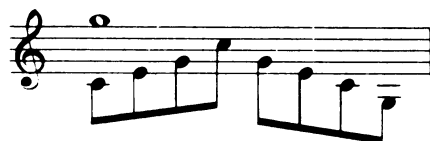


## 13/ Miscellaneous Programming Examples

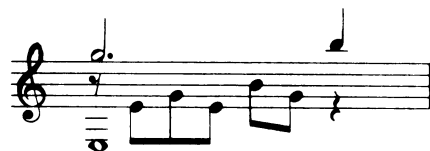
The following Music Samples are included to further educate you in the art of converting sheet music to Orchestra-90 programmed music. The key signatures are not included in the code given. The samples are in 4/4 time.



**\*I6Q4S5'6'15'Q.5**  
**@V2H52**



**\*WB**  
**V2I0247420-3**



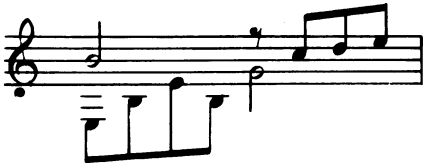
**\*H.BQD**  
**V2IS24264QS**  
**V3W-5**



**\*I.\$S7'7'I7'S7'I8Q6S8'8'**  
**V2Q5'54%4**  
**V3Q2'2-1'-1**  
**@V4Q2'25'5**



**\*TA'9IAQAAS9'7#'8#'9'**  
**V2H1#0#**



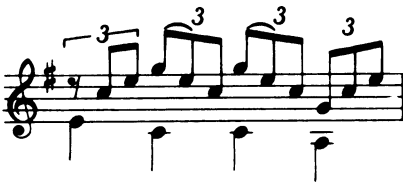
\*H6I\$789  
V2I-5-12-1H4



\*S\$3'6&'8'B'8'6'3'\$4'6&'7'I:945  
V2H-30



\*I97'Q7I9753  
V2Q5'54%'4  
V3Q3'3'3'0



\*I:\$7'9'B9'7'B9'7'4'7'9'  
V2Q20'0-2



\*ISQ9'I9\$QB'IB  
V2ISQ7'I7\$Q7#'I7  
V3ISQ4'I4\$Q6&'I6  
V4H0Q2'2



\*Q\$765  
V2H0Q23





\*ISSC%'B'I.CSB'I.CSB'D%'I.B  
V2H\$Q6%\$  
V3H1Q4

## Percussion Samples

The following sample Voicing lines demonstrate some of Orchestra-90's possible percussion effects. When arranging a piece, you would enter one for each voice you wanted as a percussion instrument. The Instrument Registration line would be entered at the very beginning of the music file. All of the following examples are labeled for Voice 4 (V4) but could be applied to any voice number. (The following is not a song, but individual program lines used to demonstrate the various effects available with just one instrument register. Experiment with the other four instrument registers to see what effects you can invent on your own.)

JES80011001E (defines Register F, Sine Wave, Volume E)  
V4YE (assigns Register F to Voice 4)  
NQ=60

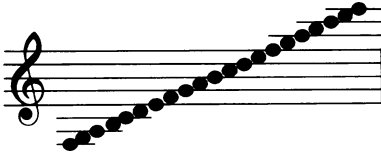
### 4/4 Time Signature

\_V4SA"B"C"D"ID"8"QD"IB"A"  
\_V4QD"E"8"IE"E"  
\_V4QC"9"IF"D"E"9"  
\_V4IE"SF"C"ID"E"SF"C"IF"D"SF"C"  
\_V4QC"F"I8"9"F"8"E"  
\_V4QC"9"D"IA"E"  
\_V4QE"I9"F"Q7"I8"9"  
\_V4QB"IA"8"E"8"6"C"  
\_V4Q8"I8"8"9"9"9"  
\_V4Q8"I9"D"8"D"A"  
\_V4Q8"SA"A"I9"F"SF"8"I8"E"SA"C"

### 3/4 Time Signature

\_V4I8"8"QC"ID"E"  
\_V4IE"SE"B"I8"B"QD"  
\_V4(ID"SE"E"ID")1  
\_V456"7"8"9"IE"8"E"8"  
\_V4S:(E"F"F")3IF"F"  
\_V4I8"8"B"S:C"D"E"ID"E"





## 14/ Using Orchestra-90 with CompuServe

Subscribing to the CompuServe Information Service gives you opportunity to be a member of the Orchestra-90 Special Interest Group. Model I/III Videotex Plus software (Catalog Number 26-1588) is available at your Radio Shack Computer Center. The Orchestra-90 Special Interest Group (SIG) gives you access to 300 available music files (in ASCII format) stored in a data base.

### Accessing and Using CompuServe's Orchestra-90 SIG

1. After you have logged into CompuServe's system, you may access the Orchestra-90 Special Interest Group by typing:

**G HOM-5** **[ENTER]**

2. After logging into the Orchestra-90 SIG, you can access the database by typing:

**XA2** **[ENTER]**

3. Then, to obtain a list of the music files, type:

**CAT /DES** **[ENTER]**

The file names will then display with brief descriptions of all files in the database. Orchestra-90 music files have the extension .A85, indicating that the files are in ASCII format and play on Orchestra-90.

### Downloading Music Files

After selecting a music file from the database, you may download it and save it on disk. For example, the command to download the file CARMEN.A85 would be:

**DOW CARMEN.A85** **[ENTER]**

CompuServe will automatically open your terminal buffer and transmit the file to your system. At the end of the file, CompuServe will close your terminal buffer. Save the file to your disk with the extension /ASC. The CARMEN. A85 file would be saved as CARMEN/ASC. This will indicate to the ORCUTL/CMD conversion utility that the music file is in ASCII format. You may now download other music files or log off the system.

### Converting ASCII Music Files

The ORCHUTIL program will allow you to convert music files from ASCII format to Orchestra 90 format.

1. At TRSDOS Ready, type:

**ORCUTL** **[ENTER]**

The ORCHUTIL module will load. The display will indicate that both the Input and Output of the program is set for /ORC.

2. To convert /ASC files to /ORC files, you must set the Input to /ASC files.

- Press **I**.

3. So that the system will READ ASCII files, you then select option A: Ascii format disk (/ASC).  
Press **A**.
4. Next, you must READ the ASCII file into memory.
  - Press **R**. The prompt displays:  
Enter INPUT file name -> \_
5. Type the file name and press **ENTER**.
6. After the file is READ into memory, you can WRITE the file to diskette. The ASCII file will be converted to a format readable by Orchestra-90 (a /ORC file).
  - Type **W**. The prompt displays:  
Enter OUTPUT file name -> \_
7. Type the file name as you want to save it and press **ENTER** (If you have more than one disk drive, you can save the file to a specified disk drive by following the file name with a: and the disk drive's number.)

You can then Play or Edit the music file.

## Appendix — Command Summary



---

CONTROL KEY	FUNCTION
<b>BREAK</b>	Return to <b>command mode</b> .
<b>CLEAR</b>	Erase to end of line.
<b>Shift BREAK</b>	Enter <b>EDIT MODE</b>
<b>Shift CLEAR</b>	Delete current line.
<b>ENTER</b>	Accept the current line.
<b>Left Arrow</b>	Move cursor left.
<b>Right Arrow</b>	Move cursor right.
<b>Shift Left Arrow</b>	Delete character.
<b>Shift Right Arrow</b>	Insert space.
<b>Up Arrow</b>	Move cursor up.
<b>Down Arrow</b>	Move cursor down.
<b>Shift Up Arrow</b>	Join line.
<b>Shift Down Arrow</b>	New line.
<b>Question Mark</b>	Duplicate line.
<b>Exclamation Mark</b>	Move line up.








---





COMMAND	OPERAND	FUNCTION
/	<any string>	Search forward for next occurrence of string. (Auto-repeat)
-	<any string>	Search backward for next occurrence of string. (Auto-repeat)
<b>APPEND</b>	<b>filename</b>	Append the named file into memory at the beginning of the current file.
<b>BOTTOM</b>	(none)	Position file pointer at the end of the file.
<b>DIR</b>	<b>0, 1, 2, or 3</b>	List all <b>/ORC</b> files on the disk named in the operand. Default= <b>0</b> .
<b>EDIT</b>	(none)	Enter <b>Edit mode</b> .
<b>GET</b>	<b>filename ...</b>	<b>Read, Score</b> and <b>Play</b> each of the files named in the operand.
<b>KILL</b>	<b>filename</b>	Delete the named file from the disk.
<b>LIST</b>	(none)	Print the current file on the printer.
<b>MULTI</b>	<b>filename...</b>	Perpetual <b>GET</b> .
<b>NEW</b>	(none)	Erase current file.
<b>PLAY</b>	<b>Part number</b> or <b>blank</b>	Play the current piece starting with the part number in the operand or at the beginning if no operand.
<b>QUIT</b>	(none)	Return control to <b>TRSDOS</b> or <b>LEVEL II</b>
<b>READ</b>	<b>filename</b>	Read the named file into memory.
<b>SCORE</b>	(none)	Compile the current file.
<b>TOP</b>	(none)	Position the text pointer at the beginning of the file.
<b>WRITE</b>	<b>filename</b>	Write the current file with the name in the operand.






















---

TIME VALUE MODIFIER	MUSICAL EQUIVALENT	NAME
. (period)		DOTTED note
: (colon)		TRIPLET

SYMBOL	MODIFIER	DEFINITION
(	(none)	Begin REITERATION.
)	hex digit	REITERATE the number of times specified by the modifier.
*	(none)	Unless otherwise indicated, all the NOTES following are assumed to be + or TREBLE clef.
/	(none)	Comments. The remainder of the line is ignored.
<	hex digit	All the NOTES in the current and following MEASURES are transposed DOWN the number of semi-tones specified by the modifier.
=	2-digit hex	Set TEMPO to value of modifier.
>	hex digit	All the NOTES in the current and following MEASURES are transposed UP the number of semi-tones specified by the modifier.
@	(none)	Unless otherwise indicated, all the NOTES following are assumed to be - or BASS clef.
_	(none)	All the unsigned NOTES following are assumed to be PERCUSSION clef. This is the underline character ( _ )
J	register name. R or S, nine hex digits	Define the named REGISTER with the named wave-form type, description and volume.
K	digit 0-7 and # or &.	Define KEY SIGNATURE by number and type specified by the modifier.
M	any string and a space	Define the beginning of a MEASURE. The current MEASURE is ended. ACCIDENTALS are dropped. the KEY SIGNATURE is restored.
N	H, Q, I, S or T	Define TIME SIGNATURE. set NOTE type in modifier to one BEAT
O	hex digit	Set the OPTIONS to the value of the modifier.
P	2-digit hex	Defines the beginning of the PART named by the modifier. The current MEASURE is ended. The current PART is ended.
R	2-digit hex	REPEAT the PART named by the modifier. The named PART must be previously defined. This symbol group may be followed by a TEMPO group and/or REGISTER group. A MEASURE following a REPEAT must be defined by a new PART number.
U	signed hex	TRANSPOSE all the NOTES following that belong to the current VOICE up or down the number of whole steps indicated by the modifier.
V	1, 2, 3, 4 or 5	All the NOTES following are added to the previous NOTES of this MEASURE belonging to the VOICE named by the modifier.
Y	A, B, C, D or E	Set the current VOICE in the current PART to the REGISTER specified by the modifier.
Z	hex digit	Map the stereo channels as specified by the modifier.

NOTE MODIFIER	MUSICAL EXAMPLE	NAME
#		ACCIDENTAL SHARP
&		ACCIDENTAL FLAT
%		ACCIDENTAL NATURAL
##		DOUBLE SHARP
&&		DOUBLE FLAT
%#		NATURAL SHARP
%&		NATURAL FLAT


EXPRESSION MODIFIER	EXAMPLE	NAME
, (comma)	 or 	SHORT STACCATO
; (semicolon)	 or 	LONG STACCATO
' (apostrophe)	(none)	SHORT ARTICULATION
" (quotation mark)	(none)	LONG ARTICULATION

TIME VALUE SYMBOL	MUSICAL EQUIVALENT	NAME OF NOTE	REST (\$)	
w		WHOLE note		
H		HALF note		
Q		QUARTER note		
I		EIGHTH note		
S		SIXTEENTH note		
T		THIRTY-SECOND note		
X		SIXTY-FOURTH note		

## NOTE SIGNS

---

**DOTTED NOTES** A dot after a note adds one-half the value of the preceding note. Examples in 4/4 time:

**Dotted Quarter Note**  = 1-1/2 beats

**Dotted Half Note**  = 3 beats

**TRIPLET NOTES** Three notes of equal time value played in the time of two of the same kinds of notes.



**HOLD**



Notes sound longer than their actual values.

**TIED NOTES**



The first of two tied notes must have no articulation. The two notes sound as one continuous note.

An accidental on the first note is passed to the note in the next measure.

**STACCATO**



Expression modifier used to make a note sound separate or detached from the other notes. Use a comma or a semicolon after the note symbol to produce this effect.

**ARTICULATION**

Expression modifier used to make a note sound clear and distinct, especially within a series of similar notes. Use an apostrophe or quotation mark after the note symbol.

---

## PERFORMANCE SIGNS

---



Repeat Sign




First and second endings




Repeat preceding measure



Common time 



Cut time 

*rit.*

Gradually slower

*Fine*

Finish

*D.C. al Fine*

Repeat from the beginning to the end of the measure marked *Fine*



- 
- D.S. al Fine* Repeat from the sign  $\text{ℌ}$  to the end of the measure marked *Fine*
- D.C. al ⊕ Coda* Repeat from the beginning to the *Coda* sign  $\oplus$  and then skip to the *Coda*
- D.S. al ⊕ Coda* Repeat from the sign  $\text{ℌ}$  to the *Coda* sign  $\oplus$  and then skip to the *Coda*
- $\text{ℌ}$  Repeat from this sign
- $\oplus$  *Coda* mark when playing second time after *D.C.* skip from this sign to the *Coda*





**RADIO SHACK, A DIVISION OF TANDY CORPORATION**

**U.S.A.: FORT WORTH, TEXAS 76102**  
**CANADA: BARRIE, ONTARIO L4M 4W5**

---

**TANDY CORPORATION**

---

**AUSTRALIA**

91 KURRAJONG ROAD  
MOUNT DRUITT, N.S.W. 2770

---

**BELGIUM**

PARC INDUSTRIEL DE NANINNE  
5140 NANINNE

---

**U.K.**

BILSTON ROAD WEDNESBURY  
WEST MIDLANDS WS10 7JN