

To: MTB Distribution
From: Paul W. Benjamin
Date: 05/13/80
Subject: Introduction of Comments to listin Segments

INTRODUCTION

For some time, lister users have been asking for the ability to put comments into their listin segments. Most have requested pll-style comments but some users indicate that they use "/"* and "*/" in their data. What follows is a description of a proposed implementation that will hopefully satisfy both groups.

DESCRIPTION

Comments will, by default, not be enabled. This is for compatibility purposes. If comments were on by default, then any application that used whatever was chosen as the default comment delimiter somewhere in its data would have to be changed. Comments will be enabled by the inclusion of the Comment_delimiter statement in the header of the listin segment. No comments will be allowed to precede this statement. It shall have the same format as the Record_delimiter and Field_delimiter statements with the extension that the characters "pll" when specified indicate that comments will begin with "/"* and end with "*/". Otherwise one of the delimiters allowed for records or fields must be supplied and that delimiter will both begin and end a comment. It must be noted that "*" is a allowable delimiter for records and fields and that care must be taken when using that as either Record_delimiter or Field_delimiter in conjunction with the usage of pll-style comments.

Comments will then be allowed between tokens or anywhere in the data. If an unpaired comment-delimiter is encountered, create_list will issue an error message with an indication as to what line the unpaired delimiter is on.

Multics Project internal working documentation. Not to be reproduced or distributed outside the Multics Project.

patient to visit the office for a checkup. Perhaps the dentist sends this letter to those selected patients who have not visited the office for six months or more.

On Multics, list processing is done with a set of commands that maintain and process online lists of information. These commands can be used to produce simple reports like the ones described above; they also provide a means by which the output can be saved in a segment or directed to the terminal. Once in a segment, the output can be mailed to other users using the Multics mail facility (see Section 8), or can be further processed by the compose command (see Section 4 of this manual) to produce reports and form letters of the highest quality.

LIST PROCESSING FILES

List processing uses three types of files, each type identified by its entryname suffix. A description of each of these types is given under the related heading below.

Listin File

A listin file is an ASCII file used to input and update a list. It is identified by the entryname suffix "listin" (e.g., monthly.listin). Records can be added to, deleted from, or updated in this list simply by editing this file with a text editor.

The format of a listin file is simple. It consists of the following three parts:

1. Header

The header specifies the record and field delimiter characters, the optional comment delimiter, and the field names. It is located at the beginning of the file and contains the following statements:

```
Comment_delimiter: c;           or      Cd; c;
Record_delimiter: r;           or      Rd: r;
Field_delimiter: f;           or      Fd: f;
Field_names: fn1, ... fnK;     or      Fn: fn1, ...fnK;
Records:
```

The Comment_delimiter statement enables comments in the listin segment and specifies the character or characters used to begin and end comments. Comments may appear anywhere that white-space is allowed and are ignored. No comments may precede this statement in the listin file. If this statement is not given, then no comments are allowed in the listin file. If the string "pll" is specified in the Comment_delimiter statement

then comments begin with "/*" and end with "*/", otherwise the comment delimiter must be exactly one character long and must be chosen from the set below, in which case that single delimiter both begins and ends comments. It should be noted that the usage of the pll-style comments in conjunction with the usage of the "*" as either record or field delimiters can cause problems and should either be avoided or used with great care. The Record_delimiter statement specifies the character used to separate records. If this statement is not given, the default record delimiter is a dollar sign (\$). The Field_delimiter statement specifies the character used to separate fields within a record. If this statement is not given, the default field delimiter character is an equal sign (=). Record and field delimiters must be exactly one character long, cannot be the same, and must be chosen from the following set:

! # \$ % & * = ? @ ^ | ~

The Records statement must be the last statement in the header, and is required. It specifies the end of the header and the beginning of the record information.

2. Fields

Fields contain the various types of information stored in a list (e.g., first name, last name, street address, date of employment, etc.). Because data records are stored separately within a listin file, the field names must be given with each data record. Within an individual data record, a field is specified by a field delimiter character followed immediately by the field name (e.g., =lname). Any amount of white space (space, horizontal tab, vertical tab, newline, or new page) can follow the field name (e.g., =lname Smith). If the field value contains any record or field delimiters, then it must be quoted (e.g., =amount "\$1.00") and in that case any embedded quotes must be doubled. A field value ends at the next record or field delimiter. Leading and trailing white space is removed from field values, though such space can be retained by including it within quotation marks that surround the entire field value.

All the field names to be used in a certain list must be specified in the Field_names statement of the header. Field names may be from 1 to 32 characters in length, must begin with an alphabetic character, and must contain only alphabetic, numeric, and underscore characters.

3. Data Records

Data records contain the specific information associated with the subject of each record. (Hereafter