To: Distribution

From: David M. Jordan

Date: December 25, 1975

Subject: Answering Service Changes for Auto Call

I. Introduction

Auto Call Units (ACUs) and their associated hardware allow a computer to place a call to another computer over standard telephone lines. This allows a relatively inexpensive alternative to more formal networks for special purpose low speed communications. Because some users are interested in using such a facility for the transfer of data, changes are being made to the Answering Service, the Ring 0 tty_ dim, and the DataNet 355 core image to implement a controlled auto call facility on Multics.

The various changes required for the tty_ dim and the DataNet 355 core image have been detailed elsewhere, so the following discussion concerns itself with the Answering Service changes being made to allow the controlled use of the auto call facility. These changes are aimed at provicing various administrative checks and at providing a system controlled sharing of the resources involved.

II. Design Considerations

The Answering Service modifications have been designed to operate in a manner consistant with the current dial facility, making use of much of the code supporting that facility, and to provide the following additional features and controls:

1. System controlled dialing, allowing records of all calls.
2. Checks that the user is permitted to use the facility.
3. Checks that the user is allowed the use of a particular auto call unit.
4. Checks that AIM attributes are consistant.
5. Facilities for the recovery of lines after process terminations, hangups, etc.

The most major changes involved are to the dial_ctl_ procedure. A new entry "dial_out_rq" is provided to allow user requested initiations and terminations of calls out. This entry checks that the requestor actually exists, that requestor's

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

process has the dialok flag on, that the requested tty channel is in the CDT and available for use, that the requested channel is an auto call channel, that the CDT access class for the channel is greater than or equal to the requestor's process authorization, and that the requestor is allowed the use of the requested channel. This last check (that the requestor is allowed) would best be implemented through the RCP system. However, RCP has not yet been extended to hanle tty channels, so dial_ctl_ implements the same function itself, using Access Control Segments (".acs") in the >system_control_1>rcp directory.

User communication with the Answering Service is providec by the dial_manager_ - as_request_ facility, which has been modified slightly to accomodate the new class of requests. Upon receiving and verifying a user request, the Answering Service (dial_ctl_) initiates the call through a tty_order which begins dialing the requested line. When an interrupt is received for the line, dial_ctl_ determines the success of the attempt through a tty status call, and informs the user. If the attempt was successful, accounting for connect time on the line is begun and the line is handed to the user process.

III. Other ImplementationDetails

In addition to the above mentioned changes to dial_manager_, as_request_ and dial_ctl_, changes are required to the following procedures: cv_cmf, up_cdt_, dialup_, as_init_, anc asu_.

cv_cmf and up_cdt_ will be modified to accept a new service type, "autocall", in the cdt.

dialup_ will be modified to aod a handler for the interrupt which occurs after dialing has completed and to check at process terminations for any active auto call lines.

as_init_ and asu_ will be modified to properly initialize the auto call lines at system initialization, and to properly set the various state variables in the CDT.