

To: Distribution
From: J. Brown
Date: May 8, 1979
Subject: ACU General Information

The interest in computer-to-computer communication has focused attention on the Automatic Call Unit (ACU) capability that is supported by Honeywell hardware and Multics. Basically, the hardware and software is in place to allow a user process to initiate a call to any node of the public dial network. Autocall requires two subchannels on the HSLA. One is used exclusively for the dialing operation, while the other is used for data transfer after the connection is established.

Both the 18X and the 355 support autocall for any communication environment that is compatible with the public network (sync/async, 110 - 4800 Baud).

The Telephone Company (TC) will provide, in addition to the telephone line, an ACU (Automatic Call Unit) and a standard modem; when an ACU application is specified, the modem and ACU will be appropriately interconnected. Optionally, a handset will also be provided. ACUs come in two flavors: tone and pulse dialing models. The TC will provide the correct type for your exchange. If your exchange can operate with either dialing system, it is recommended that the tone option be used since it is faster. Multics is made aware of an autocall channel by specifying a service type of "autocall" in the CMF. The only subchannel defined is the data channel. The existence of the ACU subchannel is assumed and the location is established by hardware. Both data and dialing commands are directed to the data channel. Hardware redirects dialing info to the associated ACU subchannel.

The user interface is provided by dial manager to allow the initiation and termination of calls. The use of autocall is controlled by the "DIALOK" attribute in the PDT and further requires "rw" permission to the ACS (Access Control Segment) for the specific autocall channel(s). (e.g., rw b.h020.acs *.Multics.*).

It should be pointed out that

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

1. All autocall channels are collected into a hunt group by `dial_manager_`.
2. `Dial_manager_` will look for the first available channel (not in use) and assign it to the user. However a specific channel may be specified by (e.g., b.h020) when calling `dial_manager_`.

If non-homogeneous autocall channels (baud rate, mode) are identified, the user should specify which channel to use. `Dial_manager` is not sensitive and will place the call regardless of compatibility.

There is a set of user procedures available on System "M" that are designed for interactive communication with foreign hosts. The set was provided by MIT and includes the interface to `dial_manager` and four called procedures to customize the data interface to various operating systems.

These are contained in the directory: `>udd>m>JBrown>mylib>s`
This program provides a full-duplex channel to a remote computer via the autocall facility.

```
dial_out {-control_args}
```

where:

- to destination
this required control argument specifies the name of a program which handles input and output for the line to the remote computer, as well as special requests.
- no phone no
telephone number to be dialed. This argument must be specified unless the input and output handler for destination supplies a default number. If specified it overrides any default. This string may be up to 20 characters. Non-numeric will be removed to form the dialed number.
- init_str, -is string
transmit string to the remote computer as soon as the dialed connection is made.
- escape, -esc <character>
make the escape character <character>. The default escape character is "!".

-chn <channel name>
 is optional and specifies a particular channel to assign. If not specified then the first available autocall channel will be assigned.

After the connection is established, all input from the remote computer is typed on user_output, and all input from user_input is transmitted to the remote computer, unless preceded by an escape character. Input preceded by the escape character is taken to be a special request. The following special requests are provided by dial_out itself. (Others may be provided by the i/o handler for the particular remote computer.) The requests are shown preceded by the default escape, "!".

!escape, !esc <character>
 change the escape character to <character>.

!execute, !e <multics command line>
 pass the string <multics command line> to the command processor.

!quit, !q
 return to Multics command level. If the request is followed by "hold" or "hd", the autocall line is left connected to the remote computer; otherwise it is hung up.

In the directory >udd>m>JBrown>mylib>s are also four called procedures that can be specified in the "-to" argument of "dial_out"

Mx: for an interactive session with a remote multics host

GCOS: same for GCOS

PRIME: same for "PRIMUS"

OS: same for IBM/TSO

It should be pointed out that a user process has no way of generating a break (quit) to the foreign system. This can be inconvenient when unwanted output is being printed. Caution can usually avoid these situations. CISL is looking at the possibility of system generated breaks.

Care should be exercised when doing blind data transfers, (file attachment), to insure the integrity of the file. Comparing the bit count in the files in questions provide reasonable assurance that the transfer has been accomplished correctly.

If the ACU channel is a 300 baud connection effective rates of >250 baud can be achieved. Using 1200 baud for dialup connection yields an effective rate of >950 baud.

The above figures were obtained using file attachment to the user i/o switch.

ACU SUBCHANNEL:

On the dual ACU subchannel (18X) are two HEX switches to associate the ACU with an adjacent data channel. This data channel must be on the same HMLC (mother) board as the ACU.

ACU

There are various options that the TC will require that the customer specify. For Multics the options should be:

- 1) Y loop start
- 2) X beginning of answer tone
- 3) T 2225 HZ answer tone
- 4) B answer detect or EON
- 5) Q data mode by contact
- 6) ZP CLR signal to data set
- 7) G terminate via data set
- 8) R stop timer on DSS on
- 9) ZT 57 sec ACR

The following is an example of a call placed from a System "M" process to Oakland University. In this example an ASCII segment is transferred from M to Oakland. Input to the local computer is designated by *LC; input to the remote computer is designated by *RC. Comments to examples are identified as /* comment */

```
*LC      pwd
         >user dir dir>Multics>JBrown
         r 1018 0.040 0.076 3

*LC      dialolk
         /* b dialolk dial out -to mx -no 8-313-377-4300 */
         channel b.h020 dialed to mx at 8-313-377-4300

         Multics MR6.5a: Oakland University, Rochester MI
         Load = 42.0 out of 90.0 units: users = 42
*RC      l JBrown
         Password:
         XXXXXXXXXXected from preemption until 1318.
         JBrown SysMaint logged in 04/10/79 1218.3 est Tue from ASCII terminal "none".
         Last login 04/05/79 1038.6 est Thu from ASCII terminal "none".
         stty -modes ^echoplex
         mail
         No mail.
         am
         ab
         cis
         No changed info segs.
         r 1218 5.478 22.715 173

*RC      pwd
         >user dir dir>SysMaint>JBrown
         r 1218 0.092 0.272 16
```

```
*LC      !e pwd      /*note use of ! to escape to local command processor*/
>user_dir_dir>Multics>JBrown
*LC      !e :xf bstat.pl1
          /* b :xf      ec >udd>m>JBrown>mylib>xferfiles */
console input : End of information reached. File input exhausted; returning to console input
r 1223 2.504 1.150 48

*RC      pwd
>user_dir_dir>SysMaint>JBrown
r 1224 0.041 0.450 21

*RC      ls bstat.**

Segments = 1, Lengths = 2.

r w    2  bstat.pl1.xfer

r 1225 0.189 1.056 32

*RC      logout
JBrown SysMaint logged out 04/10/79 1226.1 est Tue
CPU usage 9 sec, memory usage 33.1 units.
hangup

HANGUP
r 1027 111.543 8.808 297
```

mylib>xferfiles.ec

04/10/79 1028.8 mst Tue

```
&command_line off
dl start.xfer
dl term.xfer
cr start.xfer
cr term.xfer
add_line start.xfer qx
add_line start.xfer i
add_line term.xfer \f
add_line term.xfer w [segs &1].xfer
add_line term.xfer q
file_input [path start.xfer] [path &1] [path term.xfer]
&com
```