Manual: FDO91 Manual
Chapter 14: Communications Control Language (CCL) Protocol defines CCL protocol atoms and provides information for the CCL.
Last updated: January 1994

CHAPTER 14
Communications Control Language (CCL) Protocol

The Communications Control Language protocol (protocol ID 34) connects the online service client software to the remote host system. This procedure includes control over the command strings sent to the modem and the result codes expected from the modem in response. Also, once a connection is made to the modem, the Communications Control Language (CCL) controls the connection to the online service host via the appropriate data-switch networks (i.e., SprintNet, Tymnet, Datapac). The CCL module does not handle any P3 protocol communications.

CCL Module Concepts

Modem personality modules (MPMs) contain information for commanding specific brands of modems. This information includes commands to be sent to perform various functions, lists of result_strings and their meanings, and some miscellaneous advanced options, such as information on what characters may legally be dialed, and whether the modem supports error correction and flow control.

Locality groups are used to manage groups of information required for dialing to the remote host. This information includes phone numbers, baud rates, and network types (each for both a first and second attempt), phone type (DTMF or pulse), and whether or not outside-line and/or call-waiting prefixes are required, and what such prefixes are. Locality groups are intended for the convenience of users who travel with portable computers, permitting them to maintain lists of dialing information for cities they frequent. Locality groups are not referenced at the actual time of connection; however, CCL atoms are provided to move locality information to and from the member's current configuration.

CCL scripts are the driving force of the CCL mechanism. They use a simple CCL language, similar to the macro language of many terminal programs, to instruct the AOL software to properly access the various supported data-switch networks. At this time CCL scripts are provided for SprintNet (nee Telenet), Tymnet, and Datapac.

CCL Protocol Atoms

The CCL protocol atoms are described in alphabetical order in the rest of this chapter.

atom$ccl_dial

0 ($00)

Description

atom$ccl_dial is used to tell the system to begin dialing the host system. The atom references a database record to determine which network (SprintNet, Tymnet, Datapac) and, by extension, which CCL script to use to connect. The CCL script handles dialing the modem (using the currently selected modem personality module), connecting to the data-switch network, and connecting to the correct host address. On successful return, the system is in a state ready to initiate P3 protocol; on failure, an appropriate error code is returned.

Syntax

atom$ccl_dial

Return Value

A reflection of the status of the connection - zero (0) if the connection was successful; otherwise, an error code.

Example

atom$ccl_dial

atom$ccl_get_locality

4 ($04)

Description

atom$ccl_get_locality is used to retrieve setup data associated with a particular locality group. Specifically, given the global ID of a form containing a locality list of relative ID 16, it copies the setup data from the currently selected group into the database records representing the user's current configuration. This serves to select phone numbers, baud rates, networks, phone type (DTMF/pulse), and dialing prefixes.

Syntax

atom$ccl_get_locality [<global_ID>]

<global_ID> Specifies the optional global ID of the form containing locality list. If this argument is omitted, the code defaults to GLOBALID_SETUP, the top-most setup form. In any case, the specified form must contain a listbox with a relative ID of 16, and this listbox must contain a list of the available locality groups.

Return Value

Unchanged; the locality data is transparently copied into the correct database records.

Example

atom$ccl_get_locality

atom$ccl_hang_up

1 ($01)

Description

atom$ccl_hang_up tells the modem to hang up.

Syntax

```
atom$ccl_hang_up
```

Return Value

Unchanged

Example

```
atom$ccl_hang_up
```

atom$ccl_install_modem_driver
2 ($02)

Description

atom$ccl_install_modem_driver tells the CCL engine to use a new modem personality module (MPM) configuration file. All command strings, result code strings, and other data items pertinent to the new modem driver are loaded into CCL's working space. These settings remain in effect until another modem driver is installed.

Syntax

```
atom$ccl_install_modem_driver  <filename>
```

<filename>	Specifies an ASCII string (null-terminated) representing the filename of the MPM file to load. The full pathname need not be included (MPMs are assumed to be located in the subdirectory MPM\ of the online application directory), but a complete filename, including extension, must be specified.

Return Value

Unchanged

Example

```
atom$ccl_install_modem_driver  "Hayes.MDM"
```

atom$ccl_new_location
5 ($05)

Description

atom$ccl_new_location is no longer actually used by the online software, but is still supported. It creates a new locality group under a specified name, initializing the group with the current dial information. It also selects the new group as the current locality. Note that it is possible to create a duplicate group using this atom; it is the programmer's responsibility to ensure this does not happen.

Syntax

atom$ccl_new_location <group_name>

<group_name>	Specifies an ASCII string (null-terminated) representing the name of the group to create.

Return Value

Unchanged

Example

atom$ccl_new_location "Untitled"

atom$ccl_translate_baud_rate

7 ($07)

Description

atom$ccl_translate_baud_rate is designed to shorten lengthy atom streams. Given a CCL baud rate value, this atom returns an ASCII string representing the actual corresponding bps rate.

Syntax

atom$ccl_translate_baud_rate <baud_rate>

<baud_rate>	Specifies the numeric CCL baud rate value. Known CCL baud rate values are 1: 300 bps, 2: 1200 bps, 3: 2400 bps, 4: 4800 bps, 5: 9600 bps.

Return Value

The ASCII string (null-terminated) representing the baud rate specified.

Example

atom$act_get_db_value <20-0-18>
atom$uni_use_last_atom_value <prot$ccl> atom$ccl_translate_baud_rate

atom$ccl_translate_com_port
6 ($06)

Description

atom$ccl_translate_com_port is designed to shorten lengthy atom streams. Given a COM port number, this atom returns an ASCII string representing the actual corresponding port name.

Syntax

atom$ccl_translate_com_port <COM_port>

<COM_port>	Specifies the numeric COM port number. Under the MS-DOS environment, supported ports are COM1, COM2, COM3, and COM4. This may vary on different hardware platforms.

Return Value

The ASCII string (null-terminated) representing the COM port specified.

Example

```
atom$act_get_db_value  <20-0-21>
atom$uni_use_last_atom_value  <prot$ccl>  atom$ccl_translate_com_port
```

atom$ccl_translate_locality
9 ($09)

Description

atom$ccl_translate_locality retrieves the name of a specified locality group.

Syntax

atom$ccl_translate_locality <index>

<index>	Specifies the index (zero-based) of the locality group whose name you want to retrieve.

Return Value

The ASCII string (null-terminated) representing the name of the specified locality group.

Example

```
atom$act_get_db_value  <20-0-46>
atom$uni_use_last_atom_value  <prot$ccl>  atom$ccl_translate_locality
```

atom$ccl_translate_network

8 ($08)

Description

atom$ccl_translate_network returns an ASCII string representing the name of the specified network if given a data network ID value.

Syntax

atom$ccl_translate_network <net_ID>

<net_ID> Specifies the numeric network ID value. Known network IDs are:

1	SprintNet
2	Tymnet
3	Datapac

Return Value

The ASCII string (null-terminated) representing the name of the specified network.

Example

```
atom$act_get_db_value  <20-0-19>
atom$uni_use_last_atom_value <prot$ccl>  atom$ccl_translate_network
```

atom$ccl_update_locality

3 ($03)

Description

atom$ccl_update_locality is the preferred atom used for saving data to locality groups. Given the name of a locality group to save to, this atom writes the current dialing setup information, including phone numbers, baud rates, networks, phone type (DTMF/pulse), and dialing prefixes, into the specified locality group. If a group with the specified name already exists, it is updated with the new information; otherwise, a new group is created.

Syntax

atom$ccl_update_locality <local_grp_name>

<local_grp_name>	Specifies an ASCII string (null-terminated) representing the name of the locality group to which the dialing setup is saved.

Return Value

Unchanged.

Syntax

atom$ccl_update_locality "Home"