Tech Info Library

ABS Tech Note: SNA•ps10 Access/3270 Internals (4/93)

Article Created: 26 April 1993

TOPIC ---

This technical note discusses the SNA•ps implementation of the Apple 3270 API. This includes general architectural issues, values returned by requests and localization issues. Also included are error codes returned by SNA•ps Access/3270 which are not listed in the Apple 3270 API Programmer’s Manual.

DISCUSSION --

Introduction

SNA•ps Access/3270 is a full featured implementation of the Apple 3270 API over the SNA•ps gateway. All of the defined API requests are implemented providing LU1, LU2, and LU3 communication with the host.

Apple 3270 API clients should not generally need to know the details of the particular implementation of the API. This is because all implementations should fully adhere to the API specifications. This is true of the SNA•ps implementation, which was developed by the same team that designed the Apple 3270 API.

The API definition, however, does allow considerable variance in the form of returned values from API requests. The purpose of this technical note is to detail those returned values for the SNA•ps Access/3270 implementation of the Apple 3270 API. This knowledge is of value to 3270 API developers writing code to specifically handle the SNA•ps implementation of the API.

Overview

The SNA•ps Access/3270 file contains the access method code, the host connection code, the INIT code, (which loads the host connection) and a number of miscellaneous resources including icons and strings. The file is located in the Extensions folder under System 7 and in the System Folder under System 6. Whereas the access method code is a code resource managed by 3270 Prep, the host connection code is implemented as a Macintosh device driver. The access method communicates with the host connection using the device manager's Control routine.

Access Method

The access method contains the following resources:

- 3270 - Contains information about the access method returned in the FindAccessMethods request:

```
methodName    - SNA•ps Gateway    // Access method name
hcPrompt      - Gateways:         // Prompt for host connection dialog
appMem        - 8000              // Amount of application heap required
version       - 0x200             // Encoded API version number
connType      - kAppleTalkEntity  // Connection type (zone name
                                  // required)
hostLangHndl  - 0                 // Place holder for handle to list of
                                  // host languages.
theICNLID     - 150               // ICN# Resource ID
theicl4ID     - 150               // icl4 Resource ID
theicl8ID     - 150               // icl8 Resource ID
theicsLID     - 150               // ics# Resource ID
theics4ID     - 150               // ics4 Resource ID
theics8ID     - 150               // ics8 Resource ID
lcrsID        - 128               // LCRS Resource ID
trtpID        - 128               // TRTP Resource ID
hlidID        - 128               // HLID Resource ID
```

- HLID - the CGCSGID and keyboard ID for the language always supported by the host connection. While the host language and keyboard ID below refer to the base English language, note that these are subject to localization:

```
hostLangID    - 0x00650025      // The CGCSGID for the host language
keyboardID    - 1               // The keyboard ID for the host language
```

- Icons - ICN#, icl4, icl8, ics#, ics4, and ics8 all with ID 150 are the icons used to represent
  the SNA•ps Gateway and are returned by FindAccessMethods.

- LCRS - the access method code resource.

- STR# - three strings lists are defined:

```
ID 2000 - contains error strings for the SNA•ps specific error codes in
          the -8500 to -8999 range.
ID 2001 - is a place holder for error strings for the general error codes
          in the -8000 to -8499 range.
ID 2002 - contains string(s) used by the access method.  Currently this
          includes only the driver name.   If you change this string, you
          must change the driver name in the DRVR resource to match it.
```

- TRTP - a resource which describes which translation resources are required for a language to be considered valid. For this implementation, a given language is considered valid if its resources are from files with any creator ('????') and there are resources of type AtoE, EtoA, PtoE, and EtoP for a given language. AtoE and EtoA stand for ASCII to EBCDIC and EBCDIC to ASCII respectively. These API defined tables translate between Apple 3270 API ASCII and the EBCDIC code page for the given language. PtoE and EtoP stand for PC ASCII to EBCDIC and EBCDIC to PC ASCII respectively. These two table implementation defined resource types are used by the file transfer requests.

Each of these request does the following:

- OpenAccessMethod - Opens the driver in the SNA•ps Access/3270 file and allocates a small block of memory for access method storage. It then passes the request to the driver for processing.

- CloseAccessMethod - reverses the action of OpenAccessMethod. The request is given to the driver for processing, and then memory is freed.

- PollSessions - this request is not passed to the driver for processing. Instead, the driver and the access method have a locked shared PollSessions request block. The driver issues PollSessions requests on behalf of every API user during its periodic processing. Any events that have occurred are indicated in this shared request block. When an application issues PollSessions, the access method code looks at the shared PollSessions request block and copies any events returned into the passed PollSessions request block. This saves the overhead of the device manager calls for this time critical and frequently called request .

- GetErrorString - Returns error strings for all errors if they are included. Currently only SNA•ps specific errors are returned.

Host Connection

The SNA•ps Access/3270 implements the host connection as a device driver loaded by an INIT at boot time. This host connection code is a linked Macintosh application (called DrvrApp) with a device driver shell. The device driver code interfaces to the device manager with DrvrApp. The INIT loads all of DrvrApp's code resources into the system heap and builds the application jump table and global area. When a request is issued by an API client, the access method issues a call to the device driver's Control procedure. This routine sets up the application environment before passing the request to DrvrApp for processing. The combination of code and memory used by the driver is approximately 180K. DrvrApp handles all of the requests not handled by the access method.

The following lists information about the SNA•ps implementation of selected requests:

- FindHostConns - The passed parameter, zoneStr, is required by this (SNA•ps Gateway) implementation. If set to nil or *, the local zone name is used. The following describes interesting parameters returned in the HostConnInfo structure:

```
  hcName           - the AppleTalk entity object field for a SNA•ps
                     Gateway.
  strangerName     - not used in this implementation.  Always set to zero
```

length string.

defaultLang - set to the base English code page. This is the same value that is contained in the HLID resource.

hcStatus - kStranger is never returned.

sessTypes - is always set to the bitwise OR of kLU1Mask, kLU3Mask, and DisplayMask. This reflects the fact that a SNA•ps gateway always supports LU1, LU2, and LU3 session types regardless of how it's configured.

structuredFields - always set to true indicating that the host connection has code to support structured fields even if a particular session may not carry structured fields.

ftSupported - always set to true indicating file transfer support. IND$FILE is the one file transfer type supported.

dbcsSupported - always set to true indicating that the host connection has code to support DBCS data even if a particular session may not carry DBCS data.

- OpenHostConn - The passed parameter, zoneStr, is required by this (SNA•ps Gateway) implementation. If set to nil or *, the local zone name is used. If you try to activate an already active gateway, an error will be returned. When trying to activate or get the status (in FindHostConns or OpenHostConn) of a gateway, the following errors can be returned:

```
kCapyPermission      -8520   // Not authorized to download gateway
kCapyBadConfigVers   -8521   // Not a proper configuration
kCapyMSizeError      -8522   // Configuration too big
kCapyTooManyLUs      -8523   // Too many LUs for your gateway package
kCapyNotEnoughSysMem -8524   // Ran out of system heap in DynamicDownload
kCapyMCPLoadNoFWD    -8525   // Couldn't find the Forwarder
kCapyMCPLoadNoMem    -8526   // Gateway failed to get memory
kCapyToManyUsers     -8527   // Maximum of number of gateway clients
                             // exceeded
```

- FindSessions - The following describes interesting parameters returned in the HostConnInfo structure:

keyPrompt - is returned as "Enter password:" but should not be used since a session key is not required.

sessTypes - if the particular session type is known (i.e. the session is bound) then a specific value will be returned here: kLU1Mask for an LU1 session, kDisplayMask for am LU2 session, and kLU3Mask for an LU3 session. If the particular session type is not known, then the configured device type is used: if the device type is printer then kLU1Mask ORed with kLU3Mask is returned. If the device type is display, then kDisplayMask is returned.

modelRequired - is always set to false. The screenEmul field in OpenSession is not required to specify a specific model type.

keyRequired - is always set to false. The sessionKey field in OpenSession is not used.

- OpenSession - you can define a DAB containing the kFieldValidation attribute, but those DAB values will always be returned as zeros. The sessionKey field is not used. On rare occasions, OpenSession will incorrectly return a kAPISessNotFound error. This happens when more than 30 seconds have passed without a response from the gateway since the gateway was sent the verb to allocated the indicated LU. This is an unusual situation which only happens if there is extreme network load or if the gateway machine is unresponsive.The default language is used in OpenSession when either a nil hostLangDef pointer is passed or when a count of zero langauges is passed (SNA•ps v1.1.1).

- PollSessions - when an API client issues this request, the access method only returns an event that had been previously posted for that client. The actual scanning of sessions and posting of events happens in the driver during periodic time. For display sessions, events are returned in this order of priority: kSessionStatusEvent, kPSUpdateEvent, kDoSendEvent or kDoReceiveEvent, and kPassThruDataEvent. For LU1 sessions, events are returned in this order of priority: kSessionStatusEvent, kLU1DataEvent, and kPassThruDataEvent. For LU3 sessions, events are returned in this order of priority: kSessionStatusEvent, kLU3DataEvent, and kPassThruDataEvent.

- SendKeys - the following control keys are not supported and will result in an error being returned from the SendKeys request: kCKTrigger, kCKExtSelect, kCKTest, kCKTextOnOff, kCKFISymSet, kCKCursPos, kCKCursBlink, kCKAltCurs, kCKMultLogicalTerm, kCKEADocOn, kCKEADocOff, kCKEAChangeFormat, kCKEAWordDelete, kCKEAWrap.

- CopyOIA - of the defined group indicator bits, the following are NOT returned: kSetupMode, kTestMode, kSubsystemReady, kExtendedSelect, kAPL, kKana, kAlpha, kText, kUpperShift, kPSSOperSelect, kPSSFieldInherit, kNonResetMachineChk, kSecurityKey, kMachineCheck, kRetry, kDevNotWorking, kDevVeryBusy, kDeviceBusy, kMinusSymbol, kNotEnoughEntered, kWrongNumber, kNumericField, kOperUnauth, kInvalidDeadKey, kMessagePending, kPartitionWait, kHardwareMismatch, kLogTermNotConfAtCU, kAutoKeyInhibit, kAppProgHasOpInpInh, kPSSSelected, kRspTimeMonitor, kPrintCodeNotCust, kPrinterMalfunction, kPrinterPrinting, kAssignPrinter, kWhatPrinter, kPrinterAssignment, kPlay, kRecord, kRecordingOverflow, kPause, and kWindowIsEnlarged.

- StartSend - the only file transfer type supported is IFIL. All other types will result in an error. kOtherHost is an unsupported value for the hostEviron field.

- StartReceive - the only file transfer type supported is IFIL. All other types will result in an error. kOtherHost is an unsupported value for the hostEviron field.

- QuerySession - returns Structured Field and Query Reply information as listed below. Returns IFIL as the only supported file transfer type. The transportType field is always returned as kDFTSNATransport. The LU type is returned as kUnknownLU before a session is bound. The replyMode field is set to kUnknownReplyMode if a session is not bound or if it's a bound LU1 session. The sfSupport support field remains at kSFSupportUnknown until a BIND has arrived at which time it is either kSFSupported or kSFNotSupported.

Structured fields processed:

Read Partition
Erase/Reset
Set Reply Mode
Destination/Origin
Outbound 3270DS
D0 Structured Fields (if file transfer active)
SCS Data (for LU1 sessions)

Query Replys issued:

Null
Usable Area
Character Sets
Implicit Partition
Color
Highlighting
Reply Modes
DDM
Field Outlining
Paper Feed Techniques
Settable Printer Characteristics
3270 IPDS
DBCS(Asia) - if DBCS supported.

Error codes

The following error codes are specific to the SNA•ps Access/3270 API implementation:

kCapy2ndOpenAM - Second OpenAccessMethod call in access method. Impossible condition.
kCapyADSPErr - ADSP error while communicating with the gateway or gateway agent
kCapyAMPError - AMP protocol error while communicating with the gateway agent.
kCapyAbortReqLogic - Internal logic error in AbortRequest
kCapyAddUserErr - Internal error opening access method (aka adding a user).
kCapyBadSessName - Illegal LU name (session name).
kCapyCAMLogic1 - Internal logic error #1. Invalid reserved field.
kCapyCAMLogic2 - Internal logic error #2. Invalid reqID field.
kCapyCallDrvrErr - Internal logic error. Nil DrvrApp entry point.
kCapyCantFindAM - Internal logic error. Returned in debug version only.
kCapyFHCLogicErr1 - Internal logic error. PLookupName call failed.
kCapyFHCLogicErr2 - Internal logic error. NBPExtract call failed.
kCapyFTIntrnlErr - Internal file transfer logic error.
kCapyGWayCommyErr - Error communicating with the gateway.
kCapyGWayConnErr - Can't establish a gateway connection.
kCapyGWayErr - Error communicating with the gateway.

```
  kCapyInvGlobs        - Internal logic error.  Access method reserved
                         field invalid.
  kCapyNoIPCQueue      - Internal logic error.  Returned in debug version
                         only.
  kCapyOpenDriverErr   - Error opening the SNA•ps Access/3270 driver.
  kCapyResErr          - Error loading resources.
  kCapyPermission      - Not authorized to download gateway
  kCapyBadConfigVers   - Configuration isn't correct
  kCapyMSizeError      - Configuration too big
  kCapyTooManyLUs      - Too many LUs for your gateway package (invalid
                         configuration)
  kCapyNotEnoughSysMem - Ran out of system heap in DynamicDownload
  kCapyMCPLoadNoFWD    - Couldn't Find The Forwarder
  kCapyMCPLoadNoMem    - Gateway failed to get memory
  kCapyToManyUsers     - Maximum of number of gateway clients exceeded
```

Localization

There are two aspects to localization of the SNA•ps implementation of Apple's 3270 API. The first, user visible strings, are contained in STR, STR#, and 3270 resources. The second is the default host language provided if the Host Translation Tables folder either doesn't exist or doesn't contain the default language resources. To change the default language from base English, you must change the HLID resource to contain the new CGCSGID and keyboard ID values, and then replace the AtoE and EtoA tables in SNA•ps Access/3270 to contain the new language tables.

Keywords: <None>

===

19960215 11:05:19.00
Tech Info Library Article Number: 11697