New Technical Notes

Macintosh

Developer Support

TokenTalk Programmer's Guide Update

Networking **M.NW.TokenTalkProgrammer**

Written by: Rich Kubota July 1992

This Technical Note presents the additions and changes to the *TokenTalk Programmer's Guide* and the latest information with regard to software development for the Apple Token Ring NB and Token Ring 4/16 NB network cards.

Introduction

With the release of the Token Ring 4/16 NB Card, the TokenTalk driver software was modified to support the Multivendor ADEV Architecture. This architecture is described in detail in M.NW.AppleTalk2, "What's New With AppleTalk Phase 2." This Tech Note presents topics specific to TokenTalk as presented in the *TokenTalk Programmer's Guide*.

The first section, "TokenTalk Prefs File," discusses changes to the `'llcp'` resources of the TokenTalk Prefs file. The Options field is now used to store the speed setting for the card and to determine whether the Early Token Release feature is to be supported.

The next section, "TokenTalk Prep Services," describes the new TokenTalk Prep utility routines for getting and setting the default and boot parameters for an Apple MCP (Macintosh Coprocessor Platform) based token ring network card.

Getting the Latest TokenTalk Software

For testing purposes, the latest version of AppleTalk and related software can be installed by using the Network Software Installer, available on the latest Developer CD, on AppleLink on the Developer Services Bulletin Board, and on the Internet through anonymous FTP to ftp.apple.com (130.43.2.3). The Network Software Installer includes all necessary components for the TokenTalk driver software including A/ROSE.

TokenTalk Prefs File

This section discusses the changes to the `'llcp'` resource and supplements the material presented in Chapter 6, "Download and Initialization" of the *Programmer's Guide*. The

primary change to the resource concerns the 'Options' field, and the increase of the default `MaxFrame` size setting. The TokenTalk Prefs file will be found in the Preferences folder in system software version 7.0 and later, and in the System Folder for version 6.0.x. The file contains the preferred Logical Link Control (LLC) initialization parameters for a card. These parameters are stored in `'llcp'` resources with resource IDs corresponding to the slot number of the smart Token Ring Card to be supported. These initialization parameters override

the default settings provided in the TokenTalk Prep file. For example, if the user selects a smart Token Ring Card in NuBus slot 9 to be used, the TokenTalk Prep routines will search for `'llcp'` resource ID 9 in the Prefs file. If the resource is not found in the Prefs file, the default `'llcp'` parameters in the Prep file will be used.

The format of the `'llcp'` is as follows:

```
typedef struct
{
   long              FAddr;            /* Functional address */
   long              GAddr;            /* Group address */
   long              Options;          /* See below */
   void              (*Listener)();    /* Pointer to SAP 0 listener, always 0 in resource */
   unsigned short    MaxFrame;         /* Maximum frame size */
   unsigned short    StationMax;       /* Maximum number of stations */
   unsigned short    BufferSize;       /* Buffer size within chipset */
   unsigned char     SAPMax;           /* Maximum number of SAPs */
   unsigned char     GSAPMax;          /* Maximum number of GSAPs */
   unsigned char     GSAPMemberMax;    /* Maximum number of members in a group SAP */
   unsigned char     TxBufs;           /* Number of buffers in transmit list */
   unsigned char     RxBufs;           /* Number of buffers in receive list */
   unsigned char     IntMsgs;          /* Number of interrupt message buffers */
   unsigned char     TimerT1_1;        /* Default group 1 response timer */
   unsigned char     TimerT2_1;        /* Default group 1 receive acknowledge timer */
   unsigned char     TimerTI_1;        /* Default group 1 inactivity timer */
   unsigned char     TimerT1_2;        /* Default group 2 response timer */
   unsigned char     TimerT2_2;        /* Default group 2 receive acknowledge timer*/
   unsigned char     TimerTI_2;        /* Default group 2 inactivity timer */
   unsigned char     TxBufMin;         /* Transmit buffers minimum */
   unsigned char     TxBufMax;         /* Transmit buffers maximum */
   unsigned char     NodeAddr[6];      /* Node address (0 to use burned-in address) */
   unsigned char     ProdID[18];       /* Product ID string */
   unsigned char     LLCName[32];      /* IPC name of this LLC (C string) */
} TRInit;
```

Options Field

TokenTalk version 2.4 makes use of the low three bits of the `Options` field as follows:

Bits 0 and 1 are used to indicate the current speed setting for the Token Ring 4/16 NB Card as follows:

4 Mbits/sec	`bit 0 -> ON,  bit 1 -> OFF`
16 Mbits/sec	`bit 0 -> OFF, bit 1 -> ON`

Bit 2 is used to flag use of the “Early Token Release” feature : `0 -> ON, 1 -> OFF`. This feature only applies to the TokenRing 4/16 NB card when operating at 16 Mbits/sec.

For speed switchable cards, the TokenTalk Phase 2 ADEV checks whether bits 0 and 1 have been set correctly when the user selects a smart card in the network cdev. If neither bit is set, the user is presented with a dialog box allowing the user to select the desired speed. The Token Ring cdev also checks these bits to determine which radio button to select for speed switchable smart cards. For the TokenTalk NB card, which is nonswitchable, the bit settings are insignificant, and the speed is assumed to be 4 Mbits/sec.

Default Maximum Frame Size

The default MaxFrame size setting in TokenTalk version 2.3 and earlier was 1536 bytes. With TokenTalk Prep version 2.4, the maximum default frame size has been increased to 4464 bytes. The maximum frame size setting limits the amount of memory set aside by the token ring chipset for each receive buffer. If on an `LLCOpenSAP` or `LLCOpenStation` call, the `MaxIField` setting is greater than the MaxFrames value specified in the `'llcp'` resource, then the MaxFrame value is used instead.

TokenTalk Prep Services

TokenTalk Prep version 2.4 provides several new services to developers for determining and setting boot time parameters, and determining whether the Token Ring Card is speed switchable. The following new selectors are implemented in the TokenTalk Prep `'ttut'` utilities. This section supplements the material presented in Chapter 6, “Download and Initialization,” of the *Programmer’s Guide.*

`TTGetDefaultParms`	10	Return default TRInit parameters.
`TTGetBootParms`	11	Return TRInit parameters to use at the next boot time.
`TTSetBootParms`	12	Save TRInit parameters to the TokenTalk Prefs file.
`TTSpeedSwitchable`	13	Determine if the token ring speed is switchable or not.

TTGetDefaultParms

The `TTGetDefaultParms` function returns a pointer to the default LLC initialization parameters from the TokenTalk Prep file `'llcp'` resource. The second parameter to this request is a mask of the slot to operate on. This mask must have only a single bit set since the default parameters for a single card can be returned. The result of this operation is a 32-bit NuBus address, a pointer to a structure of type TRInit, if nonzero. The pointer must be disposed of when finished.

This function calls a card-specific routine to fill in the address field with that of the burned-in address from the card, the Product ID string, and the LLCName as a C string. A result of 0 indicates that

1. there is no supported card in the specified slot,
2. more than one card was specified,
3. the 'llcp' resource could not be found,
4. or a memory allocation error occurred.

The following call returns the default LLC parameters for a smart Token Ring Card in NuBus slot C:

```
result = (*UtilPtr) (TTGetDefaultParms, 1<<0xC);
```

TTGetBootParms

The `TTGetBootParms` function returns a pointer to the LLC initialization parameters to be used on the next boot of the designated card. As with the `TTGetDefaultParms` function, the second parameter to this request is a mask of the slot to operate on. This mask must have only a single bit set. This function first checks whether a card-specific `'llcp'` resource for

the card exists in the TokenTalk Prefs file; otherwise the default parameters specified in the `'llcp'` resource in the Prep file are returned. The result of this operation is a 32-bit NuBus address, a pointer to a structure of type TRInit, if nonzero. The pointer must be disposed of when finished.

This function calls a card-specific routine to fill in the address field with that of the burned-in address from the card, the Product ID string, and the LLCName as a C string. A result of 0 indicates that

1. there is no supported card in the specified slot,
2. more than one card was specified,
3. an 'llcp' resource could not be found,
4. or a memory allocation error occurred.

The following call returns the default LLC parameters for a smart Token Ring Card in NuBus slot D:

```
result = (*UtilPtr) (TTGetBootParms, 1<<0xD);
```

TTSetBootParms

The `TTSetBootParms` function stores LLC initialization parameters to the TokenTalk Prefs file. The function takes as input a pointer to a structure of type `TTSetBP` as the second parameter. The structure type, `TTSetBP`, is described below. The function searches for the Prefs file in the Preferences folder or creates the file if one does not exist. It then writes the initialization parameters to an `'llcp'` resource using the slot number passed in the `TTSetBP` parameter block. If an `'llcp'` resource already exists under the ID of the slot number, the old resource is replaced by the new parameters. The function returns `TRUE` if the parameters are successfully stored, or `FALSE` if unsuccessful. Under system software version 6.0.x, the Prefs file is searched for or created in the Blessed Folder.

The newly stored LLC initialization parameters will be used the next time the card is started. To use the burned-in address on the card as the node address of the card, set all bytes of the NodeAddr field to 0.

A result of 0 indicates that

1. a memory allocation error occurred,
2. a problem occurred opening the Preferences folder or file,
3. the Prefs file could not be created,
4. or Resource Manager error occurred.

```
typedef struct
{
      long     SlotNo;     /* Slot number to save */
      TRInit   *TRPtr;     /* Pointer to a TRInit data structure described above */
} TTSetBP;
```

The following demonstrates the use of the `TTGetBootParms` utility to set the boot parameters for a smart Token Ring Card in NuBus slot E:

```
      TRInit  myTRRec;
       TTSetBp  my SetBP
       void  *SetMyTRInitRec(TRInit &myTRRec);
```

```
SetMyTRInitRec(myTRRec); /* set LLC parameters */
mySetBP.SlotNo = 0x0E;
mySetBP.TRPtr = myTRRec;
result = (*UtilPtr) (TTGetBootParms, &mySetBP, );
```

TTSpeedSwitchable

The `TTSpeedSwitchable` function returns `true` if the speed on the Token Ring Card can be switched, or `false` if not. This function does not indicate the current speed setting for the card. To determine the speed, check the `Options` field of the `'llcp'` resource. If the two lowest bits, 0 and 1, are not set, then assume that the card supports only the 4 Mbits/sec speed. The second parameter to this request is a mask of the slot to operate on. This mask must have only a single bit set since the setting for only a single card can be returned.

The following call returns the default LLC parameters for a smart Token Ring Card in NuBus slot A:

```
result = (*UtilPtr) (TTSpeedSwitchable, 1<<0xA);
```

Further Reference:

- *TokenTalk Programmer's Guide*, Apple Computer, Inc. (M0827LL/A)
- M.NW.AppleTalk2
- *Macintosh Coprocessor Platform Developer's Kit*, Apple Computer, Inc. (M0793LL/A)