

\*\*\*\*\*

FTSC FIDONET TECHNICAL STANDARDS COMMITTEE

\*\*\*\*\*

Publication: FRL-1033
Revision: 3
Title: Automatic configuration of Points in FidoNet
Author(s): Administrator
Date: 2014-12-16

-----

Status of this document

-----

This document is a Fidonet Reference Library Document (FRL)

This document preserves FSP-1016.003 which was not considered for promotion to a standard for lack of gaining widespread use.

This document is released to the public domain, and may be used, copied or modified for any purpose whatever.

===== Original document =====

\*\*\*\*\*

FTSC FIDONET TECHNICAL STANDARDS COMMITTEE

\*\*\*\*\*

Publication: FSP-1016
Revision: 3
Title: Automatic configuration of Points in FidoNet
Author: Christian von Busse, on behalf of all Points and Nodes who took part in the development of this document.
Revision Date: 20 June 2003

Status of this document

-----

This document specifies an optional Fidonet standard protocol for the Fidonet community and requests discussion and suggestions for improvements.

This document is released to the public domain and may be used, copied or modified for any purpose whatever.

## Contents

-----

1. Introduction
2. Definitions
3. Description of the communication between CDP and CDN
  - 3.1 Files to be transfered
    - 3.1.1 From CDP to CDN
    - 3.1.2 From CDN to CDP
      - 3.1.2.1 In case of an accepted automatic application
        - 3.1.2.1.1 PPPPZZZZ.CDN
        - 3.1.2.1.2 ECHOZZZZ.ZIP
        - 3.1.2.1.3 NODEZZZZ.ZIP
      - 3.1.2.2 In case of a rejected automatic application
    - 3.2 How the files are transfered
      - 3.2.1 An address for the CDP
      - 3.2.2 Transmitting the application data
  4. Procedure after the first session
    - 4.1 Determining the passwords to be used
    - 4.2 Other recommendations
  5. Optional extensions
  6. Appendix: Example piece of source code to calculate CRC16  
  - A. Author contact data
  - B. Acknowledgements
  - C. History

## 1. Introduction

-----

This document proposes a protocol which will enable new Points, without any specific knowledge about FidoNet and its technicalities, to quickly and easily establish a link to FidoNet.

The purpose is to make it as easy for everybody to participate in FidoNet as it is to access the InterNet.

This protocol was designed to be usable, or at least to be able to be made usable, with common FidoNet Node and Point software. The effort for Nodes to accept new Points this way and the effort for developers (or users) to make their Point software compliant with this protocol has been kept as low as possible.

## 2. Definitions

---

CDP: The new Point.

CDN: A Node accepting new Points in accordance with the proposed protocol. A CDN Node carries the user-defined flag CDP in the nodelist.

Function Request: (Also called Service Request). This type of file request can be used to request a dummy filename from a Node system. This file request causes an external program to be started during the current session. Certain files may be transferred back to the requesting user in the same session.

Text File: A file containing only ASCII characters between 32 and 126, but including CR (13) and LF (10).

PPPP: Whenever this is used, it stands for the temporary Point number with which the CDP makes his/her first poll to the CDN. (See para 3.2.1). The Point number is specified by four hexadecimal digits. Leading zeros must be added if necessary.

ZZZZ: Stands for the Zone for which the CDP's application shall be valid. Normally, this is the CDN's FidoNet Zone but it may differ if the CDN has multiple addresses in multiple Zones. The Zone number is specified by four hexadecimal digits. Leading zeros must be added if necessary.

## 3. Description of the communication between CDP and CDN

---

### 3.1 Files to be Transferred

---

All files to be transferred are text files.  
They contain either comments or data.  
Comment lines start with a ';' or a '#'.  
Data lines have the format:

KEY\_WORD=VALUE

... and a length of 255 characters at maximum, including the line

termination character(s) CR, LF or CR/LF.

Key words are not case sensitive and can contain spaces in non-escaped form.

There must not be spaces before or after the =.

By way of exception ascii characters > 126 may be used in the fields which will not be used for configuring either the point software or the node software, and thus not cause any problems: These fields are:

- In PPPPZZZZ.CDP: RESIDENCE
- In PPPPZZZZ.CDN: EMAIL\_ADDR, VOICE

### 3.1.1 From CDP to CDN

-----

The CDP transmits his application to the Node in a text file. The text file is named PPPPZZZZ.CDP.

PPPPZZZZ.CDP must contain the following keywords:

#### POINTNAME

The name of the point operator as used in field 5 of a nodelist, but in the format: First\_name Last\_name

#### RESIDENCE

The place where the Point lives, format: zip\_code city

The use of zip\_code is recommended, city is mandatory.

Please remember, that a zip code is not necessarily composed of 5 numeric characters and that it also can contain alphanumeric characters.

#### PNTLST\_RES

Again, the place where the Point lives. This field contains abbreviations typically used in the Pointlist for cities, format: City

It is recommended not to allow the Point to fill out this entry but to have the setup automatically generate this value from RESIDENCE.

#### VOICEPHONE

The voice phone number of the Point, in international (ISO) standard, but spaces replaced by dashes, as in the nodelist:

+--

An area\_code must only be supplied, if it is used in the point's country.

#### TEMPAKA

The temporary AKA with which the Point calls the Node during the application poll.

#### RESULT

For faster processing on the Node system, this value contains the filename of the PPPPZZZZ.CDN file that is sent back to the CDP, should his/her application be successful.

#### PW\_USABLE

The value transmitted here specifies how many different passwords the point software can make use of. The value ranges from 1 to 4.

The numbers have the following meanings:

- 1 = The Point software can make use of only one password as session password, areafix password, file ticker password and PKT password.
- 2 = The Point software can make use of one password as session password and PKT password and can make use of a second one as areafix and file ticker password.
- 3 = The Point software can make use of one password as session and PKT password, can make use of a second one as areafix password and can make use of a third one as file ticker password.
- 4 = The Point software can be configured to make use of four different passwords for the session, areafix, the transmitted PKTs and the file ticker.

#### VERSION

The implemented version of the CDP FSP/FTS. If the keyword is omitted, version 1 is assumed. With each revision of this document the version is increased by one, so that the CDN can see, which protocol is supported by the point package.

The current version is 3.

#### 3.1.2 From CDN to CDP

-----  
Depending on whether the CDN accepts the application or not, the CDP gets a different text file as an answer.

##### 3.1.2.1 In the case of an accepted automatic application

-----  
If the application is accepted, the CDN transmits three files to the

CDP.

### 3.1.2.1.1 PPPPZZZZ.CDN

---

This file must contain the following keywords:

#### POINTNUMBER

The Point number (not the complete AKA) assigned to the CDP.  
The CDN's AKA is taken from the nodelist entry carrying the CDP flag. (Also see keyword NODE\_AKA below.)

#### PASSWORD

A password for the CDP, to be used at least as session password.  
Which passwords will really be used will be defined later on.

#### AREAFIXPW

A password which can be used as areafix password.

#### TICKERPW

A password to be used as password for the file ticker.

#### PKTPW

A password to be used as PKT password.

#### AREAFIX\_NAME

The name to which areafix messages should be addressed.

#### TICKER\_NAME

The name under which the file ticker expects to receive messages.

#### NL\_FREQ

The filename under which the current nodelist can be requested from the CDN.

#### NL\_DIFF

The name of the nodelist difference files, without extension.

#### EMAIL\_ADDR

An e-Mail address of the Node, which can be used by the Point in case of questions or difficulties.

Every Node should at least be able to transmit his fidonet.org address here:

first\_last@p0.f.n.z.fidonet.org

## VOICE

The CDN's voice phone number, which can be used by the CDP in case of questions or difficulties.

The CDN can specify a one-line text here of at most 255 characters, which is displayed to the CDP later on. So the CDN could also tell the CDP that he is not available for voice calls.

## WAIT

A non-committal time in minutes, after which, the Node system should have completely processed the CDP's application. The CDP's first call with his/her real Point data should not be made before this time has passed since his/her initial call.

WAIT passes the time to wait as a signed integer. The valid values range from 0 to 32767.

## NODE\_AKA

Since version 2 of this FSP/FTS (see 3.1.1) the node may specify the AKA (not the complete AKA, but only the node number!) under which he will list and configure the point, thus which AKA the point will have to use to build his own address.

If this keyword is omitted, the CDN's AKA will be taken from the nodelist entry carrying the CDP flag.

In case the CDN system should not support all four possible passwords, the CDN will transmit identical passwords in the different fields, according to its needs. Otherwise, all four fields will be filled with different passwords.

PW\_USABLE from the PPPZZZZ.CDP will not be evaluated for this entry in order to save time during the established connection. It will only be evaluated later on, when the CDN configures the CDP on his system.

All passwords must be 5 to 8 characters long. Passwords have to be in upper case.

### 3.1.2.1.2 ECHOZZZZ.ZIP

-----

The CDN transfers a list of the echomail areas available to the Point. This list is a text file named ECHOZZZZ.LST. It has the format compliant with the \*.NA lists:

AREATAG description

AREATAG description

[...]

The description is optional. If it is given, there has to be at least one space between the AREATAG and the description. A line may be at max 80 characters long. If a description is longer than that, it has to be continued in the next line. Lines containing only a continued description have to start with at least one space.

#### 3.1.2.1.3 NODEZZZZ.ZIP

-----

Finally, the CDN transmits a current nodelist to the CDP.

NODEZZZZ.ZIP contains a 3D nodelist with at least the CDN's home region. The name of that nodelist is the name commonly used for nodelists in zone ZZZZ.

e.g. in FidoNet Zone 2 this would be NODELIST.

#### 3.1.2.2 In case of a refused automatic application

-----

If the CDN does not accept the application, for whatever reason, the Point will receive a text file named NOPOINT.CDN.

This text file should contain an explanation as to why the entry was rejected but it may also be empty.

### 3.2 How the files are transfered

-----

All these files are transmitted within an unsecure established session between the CDN's and the CDP's FidoNet mailer.

#### 3.2.1 An address for the CDP

-----

Two CDPs must be prevented from calling the CDN at the same time with the same AKA. To achieve this, a temporary AKA is needed for the application call.

This temporary AKA is formed according to the following scheme:

Zone

The CDN's home Zone

Net

The CDN's Net

Node

Node number 9999

## Pointnumber

The Point number is created as a CRC16 (upside-down-CRC16 as used for Z-Modem and the nodelist - start polynom 11021H, start value 0 without modulation, shifting to the left) checksum created with POINTNAME, RESIDENCE and VOICEPHONE from PPPPZZZZ.CDP. The values will be concatenated directly without inserting any spaces. The checksum has then to be built modulo 32768 in order to prevent Point numbers exceeding 32767. The Point number is, as all numbers in the address, specified in decimal format.

Code examples for the most commonly used programming languages can be found in para.5 below.

### 3.2.2 Transmitting the application data

---

The CDP transmits the PPPPZZZZ.CDP. In addition, he/she file-requests the following three "MAGIC" files:

1. CDPOINT, password protected: CDP-PW
2. ECHOLIST
3. NODEZZZZ

The file request of CDPOINT initiates the creation process of PPPPZZZZ.CDN on the CDN's system. The CDNs system will pick a free Point number and randomly create the necessary password via a function file-request, if the mailer cannot react directly to received files matching a certain filemask.

The file-request of CDPOINT is password protected to minimize the chance of accidentally initiating this process.

In order to minimize the online time for the CDP, the CDP is not configured online after he has delivered his data but only after the session has terminated.

The file-request of ECHOLIST causes ECHOZZZZ.ZIP to be transmitted to the CDP.

The file-request of NODEZZZZ causes NODEZZZZ.ZIP to be transmitted to the CDP.

## 4. Procedure after the first session

---

## 4.1 Determining the passwords to be used

---

The CDN can decide how many different passwords will be used by entering the same password into multiple password fields in the file PPPPZZZZ.CDN (See para. 3.1.2.1)

The Point software transmits its capabilities via the keyword PW\_USABLE in the file PPPPZZZZ.CDP (See para. 3.1.1)

When configuring the passwords, both parties can decide exactly which password will be used by combining the PW\_USABLE entry with the transmitted passwords.

The following table will make it clear which passwords will be used:

|           | These passwords will be configured with the value of the field... |          |            |           |  |
|-----------|---|----------|------------|-----------|--|
| PW_USABLE | Session-PW  | PKT-PW   | Areafix-PW | Ticker-PW |  |
| 1         | PASSWORD  | PASSWORD | PASSWORD   | PASSWORD  |  |
| 2         | PASSWORD  | PASSWORD | AREAFIXPW  | AREAFIXPW |  |
| 3         | PASSWORD  | PASSWORD | AREAFIXPW  | TICKERPW  |  |
| 4         | PASSWORD  | PKTPW    | AREAFIXPW  | TICKERPW  |  |

## 4.2 Other recommendations

---

After the initial session has terminated, the automatic configuration of the CDP should to be initiated as quickly as possible.

The new Point should be told by his/her software that his/her application is being processed now, which will take about WAIT (See para. 3.1.2.1.1) minutes.

The Point can make use of this time by reading documents about FidoNet, which the Point software should offer to him/her.

## 5. Optional extensions

---

The CDP node and point software doesn't need to support the following extensions, there is no harm if the one or the other side doesn't support these features. But for a better handling it is

recommended to support them.

## 5.1 Removing a CDP

---

If one doesn't want to be a Fido point anymore, he (his software) sends a file to the CDP node. With this information supplied in the file the node is able to remove the point automatically from his configs.

The file is named PPPPZZZZ.DEL.

PPPPZZZZ.DEL must contain the following keywords:

### POINTNUMBER

The point number (not the complete AKA) that was assigned to the CDP who wants to be removed.

### NODENUMBER

The node number of the point's AKA. This is important, because a node with multiple AKAs may have the same point number for the different node numbers.

### PASSWORD

The password of the CDP, which is at least used as session password.

### AREAFIXPW

The password which is used as areafix password.

### TICKERPW

The password that is used as password for the file ticker.

### PKTPW

The password that is used as PKT password.

The CDP node may only accept such a PPPPZZZZ.DEL file in the secure inbound and has to check if the submitted passwords in the file are identical with the ones for the given point number. After this verification the point may (automatically) be removed from the configs and all files and mail bundles for this point may be deleted. Along with the PPPPZZZZ.DEL file the point program should send a mail to Areafix and Filefix to unsubscribe all mail and file areas.

## 5.2 Additional file-request magics

---

There are three additional magics for file-request:

1. BOXLIST
2. NETZLIST
3. NODETEXT

The file-request of BOXLIST causes BOX.LST to be transmitted to the CDP.

The file-request of NETZLIST causes NETZ.LST to be transmitted to the CDP.

The file-request of NODETEXT causes NODEHTM.ZIP to be transmitted to the CDP.

The files BOX.LST and NETZ.LST both have the same structure as the ECHOZZZZ.LST (see 3.1.2.1.2).

BOX.LST contains the echomail areas that are only available locally at the node's system.

NETZ.LST contains the echomail areas that are only available regionally at the network the node belongs to (e.g. 2432.NET in 2:2432/\*).

NODEHTM.ZIP contains one or more files (but no sub directories). The node may give some actual information to the point (his contact address and phone number, for example, further infos about his system, pictures of him, invitations to meetings, ...). These information must be in html format. The index file is node.htm - so, this one is the only file that must exist in NODEHTM.ZIP. Additionally there may be further html files and jpg graphics that are linked from node.htm.

## 6. Appendix: Example pieces of source code to calculate CRC16

---

```
Function crc16_string(InString: String): Word;
{ calculate CRC16 of a string, string is passed }
Var
  CRC   : Word; { CRC16
  i     : Integer; { Index variable for loop }
```

```

Index : Byte; { Index variable for CRC calculation }
Begin
  CRC := 0; { initialize CRC }

  { calculate CRC for every character }
  For i := 1 to Length(InString) Do
  Begin
    CRC := (CRC xor (Ord(InString[i]) SHL 8));
    For Index := 1 to 8 Do
      If ((CRC and $8000) <> 0)
      Then CRC := ((CRC SHL 1) xor $1021)
      Else CRC := (CRC SHL 1)
    End;

    crc16_string := (CRC and $FFFF) { return calculated CRC16 }
  End; { crc16_string }

```

[...]

```

Writeln('CRC16 modulo 32768 (values between 0 and 32767):');
checksum := checksum mod 32768;
Writeln(angStr:12, ' HEX-CRC/16: ', numb2hex(checksum));
Writeln(angStr:12, ' DEZ-CRC/16: ', checksum);

```

#### A. Author contact data

-----  
Christian von Busse  
Fidonet: 2:240/2188

#### B. Acknowledgements

-----  
The following people (without laying claim to completeness) have participated in developing this document:

Norbert Bilek, 2:2468/9929  
Christian von Busse, 2:240/2188  
Werner Dworak, 2:2487/9504  
Markus Engmann, 2:2483/21  
Michael Haase, 2:2432/280  
Daniel Hahler, 2:2432/337  
Wolfgang Huebner, 2:2490/1906.9  
Denny Mleinek, 2:248/7310

Dirk Pokorny, 2:2426/1210.13  
Herbert Rosenau, 2:2476/493  
Tim Schattkowsky, 2:2437/70.29  
Siggi Schoenicke, 2:2426/1225.1100  
Henning Schroeer, 2:2432/265  
Ulrich Schroeter, 2:244/1120  
Monika Steinhäuser, 2:249/3110

## C. History

-----

Rev.1, 20000717 First release.

Rev.2, 20010622 Second release. Version number added to cdp file and  
the ability for nodes to specify another AKA under  
which the point will be listed  
Proposed by Michael Haase

Rev.3, 20010906 Third release.

This release contains optional extensions to the CDP  
protocol inserted as 5. These extensions create the  
possibility to automatically remove a point from the  
system and to supply the newly created point with  
some nice looking (HTML) additional information.  
Proposed by Michael Haase

=====  
===== End Original Document =====

## Contact Data

-----

FTSC Administrator  
Fidonet: 2:2/20  
E-mail: administrator@ftsc.org

## History

-----

Rev.1, 2014-12-16: Preserves FSP-1016.001  
Rev.3, 2014-12-16: Preserevs FSP-1016.003

