



FlowCollector Binary Data File Format

The FlowCollector binary format data file can be created using **gzip** compression. This reduces the amount of disk space consumed by FlowCollector. Using compressed files enhances system performance by simplifying the conversion of data. This appendix provides a printed resource for the binary data file source code. A soft copy of the binary data file source code is located in **\$NFC_DIR/include/NFC_Datafile.h**.

```
#ifndef NFC_DATAFILE_H
#define NFC_DATAFILE_H
/*
 *-----
 * $Id$
 * $Source$
 *-----
 * Definition of datafile formats.
 *
 * Binary datafile :Each binary datafiles contains a header and
 *                  a list of records.
 *
 *                  The header contains format, aggregation,
 *                  agg_version, source, period, starttime, endtime,
 *                  activetime, flows, missed, and records.
 *
 *                  Each record structure contains a set of "Keys"
 *                  and a "Values" that is specific to the
 *                  aggregation scheme being used.
 *
 * Cisco NetFlow FlowCollector 3.6
 *
 * October 1998, Anders Fung
 *
 * Copyright (c) 1998-2002 by Cisco Systems, Inc.
 * All rights reserved.
 *-----
 * $Log$
 *-----
 * $Endlog$
 */

#define LABEL_LEN      16
#define IP_LEN         15
#define ASCII_HEADER_LEN 511
#define BIN_FILE_SUFFIX ".bin"

#ifndef __NFC__
enum Aggregation
```

```

{
    noAgg,                /* reserved */
    RawFlows,             /* Not supported in binary files */
    SourceNode,
    DestNode,
    HostMatrix,
    SourcePort,
    DestPort,
    Protocol,
    DetailDestNode,
    DetailHostMatrix,
    DetailInterface,
    CallRecord,
    ASMatrix,
    NetMatrix,
    DetailSourceNode,
    DetailASMatrix,
    ASHostMatrix,
    HostMatrixInterface,
    DetailCallRecord,
    RouterAS,
    RouterProtoPort,
    RouterSrcPrefix,
    RouterDstPrefix,
    RouterPrefix,
    ASPort,
    RouterTosAS,
    RouterTosProtoPort,
    RouterTosSrcPrefix,
    RouterTosDstPrefix,
    RouterTosPrefix,
    RouterPrePortProtocol,
    RouterDestOnly,
    RouterSrcDst,
    RouterFullFlow,
    MPLSVPNUUsage
};
#endif

typedef struct {
    u_short format;        /* Header format, it is 2 in this round */
    char  newline;         /* Newline character, '\n' */
    char  ascii_header[ASCII_HEADER_LEN]; /* Header in ASCII */
    u_char aggregation;    /* Aggregation scheme used */
    u_char agg_version;    /* Version of the aggregation scheme used */
    char  source[IP_LEN];  /* Source IP/Name */
    u_char period;         /* Aggregation period, 0 means PARTIAL */
    u_long starttime;      /* Beginning of aggregation period */
    u_long endtime;        /* End of aggregation period */
    u_long flows;          /* Number of flows aggregated */
    int    missed;         /* Number of flows missed, -1 means not avail*/
    u_long records;        /* Number of records in this datafile */
} BinaryHeaderF2;

#define HEADER_FORMAT_2 2

typedef struct {
    /* Keys */
    u_long  srcaddr;        /* Source IP */

    /* Values */
    u_long  pkts;           /* Packet count */

```

```

        u_long  octets;           /* Byte count */
        u_long  flows;           /* Flow count */

} BinaryRecord_SourceNode_V1;

#define SOURCENODE_V1 1

typedef struct {
        u_long  dstaddr;         /* Keys */
                                /* Destination IP */

                                /* Values */
        u_long  pkts;           /* Packet count */
        u_long  octets;         /* Byte count */
        u_long  flows;          /* Flow count */
} BinaryRecord_DestNode_V1;

#define DESTNODE_V1 1

typedef struct {
        u_long  srcaddr;         /* Keys */
        u_long  dstaddr;         /* Source IP */
                                /* Destination IP */

                                /* Values */
        u_long  pkts;           /* Packet count */
        u_long  octets;         /* Byte count */
        u_long  flows;          /* Flow count */
} BinaryRecord_HostMatrix_V1;

#define HOSTMATRIX_V1 1

typedef struct {
        char    srcport[LABEL_LEN]; /* Keys */
                                /* Source Port Key */

                                /* Values */
        u_long  pkts;           /* Packet count */
        u_long  octets;         /* Byte count */
        u_long  flows;          /* Flow count */
} BinaryRecord_SourcePort_V1;

#define SOURCEPORT_V1 1

typedef struct {
        char    dstport[LABEL_LEN]; /* Keys */
                                /* Destination Port Key */

                                /* Values */
        u_long  pkts;           /* Packet count */
        u_long  octets;         /* Byte count */
        u_long  flows;          /* Flow count */
} BinaryRecord_DestPort_V1;

#define DESTPORT_V1 1

typedef struct {
        char    protocol[LABEL_LEN]; /* Keys */
                                /* Protocol Key */

```

```

        /* Values */
        u_long  pkts;           /* Packet count */
        u_long  octets;        /* Byte count */
        u_long  flows;         /* Flow count */
    } BinaryRecord_Protocol_V1;

#define PROTOCOL_V1 1

typedef struct {
        /* Keys */
        u_long  srcaddr;       /* Source IP */
        char    srcport[LABEL_LEN]; /* Source Port Key */
        char    dstport[LABEL_LEN]; /* Destination Port Key */
        char    protocol[LABEL_LEN]; /* Protocol Key */

        /* Values */
        u_long  pkts;           /* Packet count */
        u_long  octets;        /* Byte count */
        u_long  flows;         /* Flow count */
    } BinaryRecord_DetailSourceNode_V1;

#define DETAIL_SOURCENODE_V1 1

typedef struct {
        /* Keys */
        u_long  dstaddr;       /* Destination IP */
        char    srcport[LABEL_LEN]; /* Source Port Key */
        char    dstport[LABEL_LEN]; /* Destination Port Key */
        char    protocol[LABEL_LEN]; /* Protocol Key */

        /* Values */
        u_long  pkts;           /* Packet count */
        u_long  octets;        /* Byte count */
        u_long  flows;         /* Flow count */
    } BinaryRecord_DetailDestNode_V1;

#define DETAIL_DESTNODE_V1 1

typedef struct {
        /* Keys */
        u_long  srcaddr;       /* Source IP */
        u_long  dstaddr;       /* Destination IP */
        char    srcport[LABEL_LEN]; /* Source Port Key */
        char    dstport[LABEL_LEN]; /* Destination Port Key */
        char    protocol[LABEL_LEN]; /* Protocol Key */

        /* Values */
        u_long  pkts;           /* Packet count */
        u_long  octets;        /* Byte count */
        u_long  flows;         /* Flow count */
        u_long  starttime;     /* Start time */
        u_long  endtime;       /* End time */
    } BinaryRecord_DetailHostMatrix_V1;

#define DETAIL_HOSTMATRIX_V1 1

typedef struct {
        /* Keys */
        u_long  srcaddr;       /* Source IP */

```

```

        u_long  dstaddr;          /* Destination IP */
        u_short input;           /* Input Interface Number */
        u_short output;         /* Output Interface Number */
        u_long  nexthop;         /* Next Hop IP */

                                   /* Values */
        u_long  pkts;            /* Packet count */
        u_long  octets;          /* Byte count */
        u_long  flows;           /* Flow count */
    } BinaryRecord_DetailInterface_V1;

#define DETAIL_INTERFACE_V1 1

typedef struct {
                                   /* Keys */
        u_long  srcaddr;         /* Source IP */
        u_long  dstaddr;         /* Destination IP */
        u_short srcport;         /* Source Port Number */
        u_short dstport;         /* Destination Port Number */
        u_char  prot;            /* Protocol Number */
        u_char  tos;             /* Type of Service */
        u_short reserved;        /* Data alignment */

                                   /* Values */
        u_long  pkts;            /* Packet count */
        u_long  octets;          /* Byte count */
        u_long  flows;           /* Flow count */
        u_long  starttime;       /* Start time */
        u_long  endtime;         /* End time */
        u_long  activetime;       /* Total Active Time */
    } BinaryRecord_CallRecord_V1;

#define CALLRECORD_V1 1

typedef struct {
                                   /* Keys */
        char    src_as[LABEL_LEN]; /* Source AS */
        char    dst_as[LABEL_LEN]; /* Destination AS */

                                   /* Values */
        u_long  pkts;            /* Packet count */
        u_long  octets;          /* Byte count */
        u_long  flows;           /* Flow count */
    } BinaryRecord_ASMatrix_V1;

#define ASMATRIX_V1 1

typedef struct {
                                   /* Keys */
        u_long  srcaddr;         /* Source IP */
        u_long  dstaddr;         /* Destination IP */
        char    src_as[LABEL_LEN]; /* Source AS */
        char    dst_as[LABEL_LEN]; /* Destination AS */
        u_short input;           /* Input Interface Number */
        u_short output;          /* Output Interface Number */
        char    srcport[LABEL_LEN]; /* Source Port Key */
        char    dstport[LABEL_LEN]; /* Destination Port Key */
        char    protocol[LABEL_LEN]; /* Protocol Key */

                                   /* Values */
        u_long  pkts;            /* Packet count */

```

```

    u_long  octets;          /* Byte count */
    u_long  flows;           /* Flow count */
} BinaryRecord_DetailASMatrix_V1;

#define DETAIL_ASMATRIX_V1 1

typedef struct {
    /* Keys */
    u_long  src_subnet;      /* Source SubNet */
    u_short src_mask;        /* Source SubNet Mask */
    u_short input;           /* Input Interface Number */
    u_long  dst_subnet;      /* Destination SubNet */
    u_short dst_mask;        /* Destination SubNet Mask */
    u_short output;          /* Output Interface Number */

    /* Values */
    u_long  pkts;            /* Packet count */
    u_long  octets;          /* Byte count */
    u_long  flows;           /* Flow count */
} BinaryRecord_NetMatrix_V1;

#define NETMATRIX_V1 1

typedef struct {
    /* Keys */
    char    src_as[LABEL_LEN]; /* Source AS */
    char    dst_as[LABEL_LEN]; /* Destination AS */
    char    srcport[LABEL_LEN]; /* Source Port Key */
    char    dstport[LABEL_LEN]; /* Destination Port Key */
    char    protocol[LABEL_LEN]; /* Protocol Key */

    /* Values */
    u_long  pkts;            /* Packet count */
    u_long  octets;          /* Byte count */
    u_long  flows;           /* Flow count */
} BinaryRecord_ASPort_V1;

#define ASPORT_V1 1

typedef struct {
    /* Keys */
    char    src_as[LABEL_LEN]; /* Source AS */
    char    dst_as[LABEL_LEN]; /* Destination AS */
    u_short input;             /* Input Interface Number */
    u_short output;            /* Output Interface Number */

    /* Values */
    u_long  pkts;            /* Packet count */
    u_long  octets;          /* Byte count */
    u_long  flows;           /* Flow count */
    u_long  starttime;       /* Start time */
    u_long  endtime;         /* End time */
    u_long  activetime;       /* Total Active Time */
} BinaryRecord_RouterAS_V1;

#define ROUTERAS_V1 1

typedef struct {
    /* Keys */
    char    srcport[LABEL_LEN]; /* Source Port Key */
    char    dstport[LABEL_LEN]; /* Destination Port Key */
    u_char  prot;               /* Protocol Number */

```

```

    u_char  pad;                /* Data alignment */
    u_short reserved;           /* Data alignment */

                                /* Values */
    u_long  pkts;               /* Packet count */
    u_long  octets;             /* Byte count */
    u_long  flows;              /* Flow count */
    u_long  starttime;          /* Start time */
    u_long  endtime;            /* End time */
    u_long  activetime;         /* Total Active Time */
} BinaryRecord_RouterProtoPort_V1;

#define ROUTERPROTOPORT_V1 1

typedef struct {
                                /* Keys */
    u_long  src_subnet;         /* Source SubNet */
    u_short src_mask;           /* Source SubNet Mask */
    u_short input;              /* Input Interface Number */
    char     src_as[LABEL_LEN]; /* Source AS */

                                /* Values */
    u_long  pkts;               /* Packet count */
    u_long  octets;             /* Byte count */
    u_long  flows;              /* Flow count */
    u_long  starttime;          /* Start time */
    u_long  endtime;            /* End time */
    u_long  activetime;         /* Total Active Time */
} BinaryRecord_RouterSrcPrefix_V1;

#define ROUTERSRCPREFIX_V1 1

typedef struct {
                                /* Keys */
    u_long  dst_subnet;         /* Destination SubNet */
    u_short dst_mask;           /* Destination SubNet Mask */
    u_short output;             /* Output Interface Number */
    char     dst_as[LABEL_LEN]; /* Destination AS */

                                /* Values */
    u_long  pkts;               /* Packet count */
    u_long  octets;             /* Byte count */
    u_long  flows;              /* Flow count */
    u_long  starttime;          /* Start time */
    u_long  endtime;            /* End time */
    u_long  activetime;         /* Total Active Time */
} BinaryRecord_RouterDstPrefix_V1;

#define ROUTERDSTPREFIX_V1 1

typedef struct {
                                /* Keys */
    u_long  src_subnet;         /* Source SubNet */
    u_long  dst_subnet;         /* Destination SubNet */
    u_short src_mask;           /* Source SubNet Mask */
    u_short dst_mask;           /* Destination SubNet Mask */
    u_short input;              /* Input Interface Number */
    u_short output;             /* Output Interface Number */
    char     src_as[LABEL_LEN]; /* Source AS */
    char     dst_as[LABEL_LEN]; /* Destination AS */

```

```

        /* Values */
        u_long  pkts;           /* Packet count */
        u_long  octets;         /* Byte count */
        u_long  flows;          /* Flow count */
        u_long  starttime;      /* Start time */
        u_long  endtime;        /* End time */
        u_long  activetime;     /* Total Active Time */
    } BinaryRecord_RouterPrefix_V1;

#define ROUTERPREFIX_V1 1

typedef struct {
        /* Keys */
        char    src_as[LABEL_LEN]; /* Source AS */
        char    dst_as[LABEL_LEN]; /* Destination AS */
        u_short input;              /* Input Interface Number */
        u_short output;             /* Output Interface Number */
        u_char  tos;                /* Type of Service */

        /* Values */
        u_long  pkts;           /* Packet count */
        u_long  octets;         /* Byte count */
        u_long  flows;          /* Flow count */
        u_long  starttime;      /* Start time */
        u_long  endtime;        /* End time */
        u_long  activetime;     /* Total Active Time */
    } BinaryRecord_RouterTosAS_V1;

#define ROUTERTOSAS_V1 1

typedef struct {
        /* Keys */
        char    srcport[LABEL_LEN]; /* Source Port Key */
        char    dstport[LABEL_LEN]; /* Destination Port Key */
        u_char  prot;                /* Protocol Number */
        u_short input;              /* Input Interface Number */
        u_short output;             /* Output Interface Number */
        u_char  tos;                /* Type of Service */

        /* Values */
        u_long  pkts;           /* Packet count */
        u_long  octets;         /* Byte count */
        u_long  flows;          /* Flow count */
        u_long  starttime;      /* Start time */
        u_long  endtime;        /* End time */
        u_long  activetime;     /* Total Active Time */
    } BinaryRecord_RouterTosProtoPort_V1;

#define ROUTERTOSPROTOPORT_V1 1

typedef struct {
        /* Keys */
        u_long  src_subnet;        /* Source SubNet */
        u_short src_mask;          /* Source SubNet Mask */
        u_long  dst_subnet;        /* Destination SubNet */
        u_short dst_mask;          /* Destination SubNet Mask */
        char    srcport[LABEL_LEN]; /* Source Port Key */
        char    dstport[LABEL_LEN]; /* Destination Port Key */
        u_char  prot;                /* Protocol Number */
        u_short input;              /* Input Interface Number */
        u_short output;             /* Output Interface Number */
        u_char  tos;                /* Type of Service */

```

```

/* Values */
u_long  pkts;          /* Packet count */
u_long  octets;        /* Byte count */
u_long  flows;         /* Flow count */
u_long  starttime;     /* Start time */
u_long  endtime;       /* End time */
u_long  activetime;    /* Total Active Time */
} BinaryRecord_RouterPrePortProtocol_V1;

#define ROUTERPREPORTPROTOCOL_V1 1

typedef struct {
/* Keys */
u_long  dstaddr;       /* Destination IP */

/* Values */
u_long  pkts;          /* Packet count */
u_long  octets;        /* Byte count */
u_long  flows;         /* Flow count */
u_long  starttime;     /* Start time */
u_long  endtime;       /* End time */
u_long  activetime;    /* Total Active Time */
} BinaryRecord_RouterDestOnly_V1;

#define ROUTERDESTONLY_V1 1

typedef struct {
/* Keys */
u_long  srcaddr;       /* Source IP */
u_long  dstaddr;       /* Destination IP */

/* Values */
u_long  pkts;          /* Packet count */
u_long  octets;        /* Byte count */
u_long  flows;         /* Flow count */
u_long  starttime;     /* Start time */
u_long  endtime;       /* End time */
u_long  activetime;    /* Total Active Time */
} BinaryRecord_RouterSrcDst_V1;

#define ROUTERSRCDST_V1 1

typedef struct {
/* Keys */
u_long  srcaddr;       /* Source IP */
u_long  dstaddr;       /* Destination IP */
char    srcport[LABEL_LEN]; /* Source Port Key */
char    dstport[LABEL_LEN]; /* Destination Port Key */
u_char  prot;          /* Protocol Number */

/* Values */
u_long  pkts;          /* Packet count */
u_long  octets;        /* Byte count */
u_long  flows;         /* Flow count */
u_long  starttime;     /* Start time */
u_long  endtime;       /* End time */
u_long  activetime;    /* Total Active Time */
} BinaryRecord_RouterFullFlow_V1;

#define ROUTERFULLFLOW_V1 1

typedef struct {
/* Keys */
u_long  src_subnet;    /* Source SubNet */

```

```

    u_short src_mask;          /* Source SubNet Mask */
    u_short input;             /* Input Interface Number */
    char    src_as[LABEL_LEN]; /* Source AS */
    u_char  tos;               /* Type of Service */

                                /* Values */
    u_long  pkts;              /* Packet count */
    u_long  octets;            /* Byte count */
    u_long  flows;             /* Flow count */
    u_long  starttime;         /* Start time */
    u_long  endtime;           /* End time */
    u_long  activetime;        /* Total Active Time */
} BinaryRecord_RouterTosSrcPrefix_V1;

#define ROUTERTOSSRCPREFIX_V1 1

typedef struct {
                                /* Keys */
    u_long  dst_subnet;        /* Destination SubNet */
    u_short dst_mask;          /* Destination SubNet Mask */
    u_short output;            /* Output Interface Number */
    char    dst_as[LABEL_LEN]; /* Destination AS */
    u_char  tos;               /* Type of Service */

                                /* Values */
    u_long  pkts;              /* Packet count */
    u_long  octets;            /* Byte count */
    u_long  flows;             /* Flow count */
    u_long  starttime;         /* Start time */
    u_long  endtime;           /* End time */
    u_long  activetime;        /* Total Active Time */
} BinaryRecord_RouterTosDstPrefix_V1;

#define ROUTERTOSDSTPREFIX_V1 1

typedef struct {
                                /* Keys */
    u_long  src_subnet;        /* Source SubNet */
    u_long  dst_subnet;        /* Destination SubNet */
    u_short src_mask;          /* Source SubNet Mask */
    u_short dst_mask;          /* Destination SubNet Mask */
    u_short input;             /* Input Interface Number */
    u_short output;            /* Output Interface Number */
    char    src_as[LABEL_LEN]; /* Source AS */
    char    dst_as[LABEL_LEN]; /* Destination AS */
    u_char  tos;               /* Type of Service */

                                /* Values */
    u_long  pkts;              /* Packet count */
    u_long  octets;            /* Byte count */
    u_long  flows;             /* Flow count */
    u_long  starttime;         /* Start time */
    u_long  endtime;           /* End time */
    u_long  activetime;        /* Total Active Time */
} BinaryRecord_RouterTosPrefix_V1;

#define ROUTERTOSPFX_V1 1

typedef struct {
                                /* Keys */
    u_long  srcaddr;           /* Source IP */

```

```

    u_long  dstaddr;          /* Destination IP */
    char    src_as[LABEL_LEN]; /* Source AS */
    char    dst_as[LABEL_LEN]; /* Destination AS */

                                /* Values */
    u_long  pkts;             /* Packet count */
    u_long  octets;           /* Byte count */
    u_long  flows;            /* Flow count */
    u_long  starttime;        /* Start time */
    u_long  endtime;          /* End time */
    u_long  activetime;       /* Total Active Time */
} BinaryRecord_ASHostMatrix_V1;

#define ASHOSTMATRIX_V1 1

typedef struct {
                                /* Keys */
    u_long  srcaddr;           /* Source IP */
    u_long  dstaddr;           /* Destination IP */
    u_short input;             /* Input Interface Number */
    u_short output;            /* Output Interface Number */
    char    protocol[LABEL_LEN]; /* Protocol Key */

                                /* Values */
    u_long  pkts;             /* Packet count */
    u_long  octets;           /* Byte count */
    u_long  flows;            /* Flow count */
} BinaryRecord_HostMatrixInterface_V1;

#define HOSTMATRIXINTERFACE_V1 1

typedef struct {
                                /* Keys */
    u_long  srcaddr;           /* Source IP */
    u_long  dstaddr;           /* Destination IP */
    char    srcport[LABEL_LEN]; /* Source Port Key */
    char    dstport[LABEL_LEN]; /* Destination Port Key */
    u_short input;             /* Input Interface Number */
    u_short output;            /* Output Interface Number */
    char    protocol[LABEL_LEN]; /* Protocol Key */
    u_char  tos;                /* Type of Service */
    u_char  pad;                /* Data alignment */
    u_short reserved;          /* Data alignment */

                                /* Values */
    u_long  pkts;             /* Packet count */
    u_long  octets;           /* Byte count */
    u_long  flows;            /* Flow count */
    u_long  starttime;        /* Start time */
    u_long  endtime;          /* End time */
    u_long  activetime;       /* Total Active Time */
} BinaryRecord_DetailCallRecord_V1;

#define DETAILCALLRECORD_V1 1

typedef struct {
                                /* Keys */
    u_long  srcaddr;           /* Source IP */
    u_long  dstaddr;           /* Destination IP */
    u_short input;             /* Input Interface Number */
    u_char  tos;                /* Type of Service */

```

```

    u_char  pad;                /* Data alignment */

                                /* Values */
    u_long  pkts;               /* Packet count */
    u_long  octets;            /* Byte count */
    u_long  flows;             /* Flow count */
} BinaryRecord_InterfaceMatrix_V1;

#define INTERFACEMATRIX_V1 1

typedef struct {
                                /* Keys */
    u_long  srcaddr;           /* Source IP */
    u_long  dstaddr;           /* Destination IP */
    u_short input;             /* Input Interface Number */
    u_char  tos;               /* Type of Service */
    u_char  pad;               /* Data alignment */

                                /* Values */
    u_long  pkts;               /* Packet count */
    u_long  octets;            /* Byte count */
    u_long  flows;             /* Flow count */
    u_short custlength;        /* Length of custom data*/
    u_short reserved;          /* Data alignment */
    u_char* customer;          /* Customer ID*/
    char*   site;              /* Site ID*/
    char*   ce;                /* CE Name*/

} BinaryRecord_MPLSVPNUUsage_V1;

#define MPLSVPNUSAGE_V1 1

typedef struct {
    BinaryHeaderF2 header;
    union {
        BinaryRecord_SourceNode_V1      * srcnode;
        BinaryRecord_DestNode_V1         * dstnode;
        BinaryRecord_HostMatrix_V1       * hostmatrix;
        BinaryRecord_SourcePort_V1       * srcport;
        BinaryRecord_DestPort_V1         * dstport;
        BinaryRecord_Protocol_V1         * protocol;
        BinaryRecord_DetailSourceNode_V1 * detailsrcnode;
        BinaryRecord_DetailDestNode_V1   * detaildstnode;
        BinaryRecord_DetailHostMatrix_V1 * detailhostmatrix;
        BinaryRecord_DetailInterface_V1  * detailinterface;
        BinaryRecord_CallRecord_V1       * callrecord;
        BinaryRecord_ASMatrix_V1         * asmatrix;
        BinaryRecord_DetailASMatrix_V1   * detailasmatrix;
        BinaryRecord_NetMatrix_V1        * netmatrix;
        BinaryRecord_ASHostMatrix_V1     * ashostmatrix;
        BinaryRecord_HostMatrixInterface_V1 * hostmatrixinterface;
        BinaryRecord_DetailCallRecord_V1 * detailcallrecord;
        BinaryRecord_ASPort_V1           * asport;
        BinaryRecord_RouterAS_V1         * routeras;
        BinaryRecord_RouterTosAS_V1      * routertosas;
        BinaryRecord_RouterProtoPort_V1  * routerprotoport;
        BinaryRecord_RouterTosProtoPort_V1 * routertosprotoport;
        BinaryRecord_RouterPrePortProtocol_V1 * routerpreportprotocol;
        BinaryRecord_RouterSrcPrefix_V1  * routersrcprefix;
        BinaryRecord_RouterTosSrcPrefix_V1 * routertossrcprefix;
        BinaryRecord_RouterDstPrefix_V1  * routerdstprefix;
    };
}

```

```

        BinaryRecord_RouterTosDstPrefix_V1 * routertosdstprefix;
        BinaryRecord_RouterPrefix_V1      * routerprefix;
        BinaryRecord_RouterTosPrefix_V1   * routertosprefix;
        BinaryRecord_RouterDestOnly_V1    * routerdestonly;
        BinaryRecord_RouterSrcDst_V1       * routersrcdst;
        BinaryRecord_RouterFullFlow_V1    * routerfullflow;
        BinaryRecord_InterfaceMatrix_V1    * interfacematrix;
    } record;
} BinaryDatafile;

#define MAX_BINARY_HEADER_F2 \
    (sizeof(BinaryHeaderF2))

#define MAX_BINARY_RECORD_SOURCE_NODE_SIZE \
    (sizeof(BinaryRecord_SourceNode_V1))

#define MAX_BINARY_RECORD_DESTINATION_NODE_SIZE \
    (sizeof(BinaryRecord_DestNode_V1))

#define MAX_BINARY_RECORD_HOST_MATRIX_SIZE \
    (sizeof(BinaryRecord_HostMatrix_V1))

#define MAX_BINARY_RECORD_SOURCE_PORT_SIZE \
    (sizeof(BinaryRecord_SourcePort_V1))

#define MAX_BINARY_RECORD_DESTINATION_PORT_SIZE \
    (sizeof(BinaryRecord_DestPort_V1))

#define MAX_BINARY_RECORD_PROTOCOL_SIZE \
    (sizeof(BinaryRecord_Protocol_V1))

#define MAX_BINARY_RECORD_DETAIL_SOURCE_NODE_SIZE \
    (sizeof(BinaryRecord_DetailSourceNode_V1))

#define MAX_BINARY_RECORD_DETAIL_DESTINATION_NODE_SIZE \
    (sizeof(BinaryRecord_DetailDestNode_V1))

#define MAX_BINARY_RECORD_DETAIL_HOST_MATRIX_SIZE \
    (sizeof(BinaryRecord_DetailHostMatrix_V1))

#define MAX_BINARY_RECORD_DETAIL_INTERFACE_SIZE \
    (sizeof(BinaryRecord_DetailInterface_V1))

#define MAX_BINARY_RECORD_CALL_RECORD_SIZE \
    (sizeof(BinaryRecord_CallRecord_V1))

#define MAX_BINARY_RECORD_INTERFACE_MATRIX_SIZE \
    (sizeof(BinaryRecord_InterfaceMatrix_V1))

#define MAX_BINARY_RECORD_AS_MATRIX_SIZE \
    (sizeof(BinaryRecord_ASMatrix_V1))

#define MAX_BINARY_RECORD_DETAIL_AS_MATRIX_SIZE \
    (sizeof(BinaryRecord_DetailASMatrix_V1))

#define MAX_BINARY_RECORD_NET_MATRIX_SIZE \
    (sizeof(BinaryRecord_NetMatrix_V1))

#define MAX_BINARY_RECORD_AS_HOST_MATRIX_SIZE \
    (sizeof(BinaryRecord_ASHostMatrix_V1))

#define MAX_BINARY_RECORD_HOST_MATRIX_INTERFACE_SIZE \
    (sizeof(BinaryRecord_HostMatrixInterface_V1))

```

```

#define MAX_BINARY_RECORD_DETAIL_CALL_RECORD_SIZE \
    (sizeof(BinaryRecord_DetailCallRecord_V1))

#define MAX_BINARY_RECORD_AS_PORT_SIZE \
    (sizeof(BinaryRecord_ASPort_V1))

#define MAX_BINARY_RECORD_ROUTER_AS_SIZE \
    (sizeof(BinaryRecord_RouterAS_V1))

#define MAX_BINARY_RECORD_ROUTER_TOS_AS_SIZE \
    (sizeof(BinaryRecord_RouterTosAS_V1))

#define MAX_BINARY_RECORD_ROUTER_PROTO_PORT_SIZE \
    (sizeof(BinaryRecord_RouterProtoPort_V1))

#define MAX_BINARY_RECORD_ROUTER_TOS_PROTO_PORT_SIZE \
    (sizeof(BinaryRecord_RouterTosProtoPort_V1))

#define MAX_BINARY_RECORD_ROUTER_PRE_PORT_PROTOCOL_SIZE \
    (sizeof(BinaryRecord_RouterPrePortProtocol_V1))

#define MAX_BINARY_RECORD_ROUTER_SRC_PREFIX_SIZE \
    (sizeof(BinaryRecord_RouterSrcPrefix_V1))

#define MAX_BINARY_RECORD_ROUTER_TOS_SRC_PREFIX_SIZE \
    (sizeof(BinaryRecord_RouterTosSrcPrefix_V1))

#define MAX_BINARY_RECORD_ROUTER_DST_PREFIX_SIZE \
    (sizeof(BinaryRecord_RouterDstPrefix_V1))

#define MAX_BINARY_RECORD_ROUTER_TOS_DST_PREFIX_SIZE \
    (sizeof(BinaryRecord_RouterTosDstPrefix_V1))

#define MAX_BINARY_RECORD_ROUTER_PREFIX_SIZE \
    (sizeof(BinaryRecord_RouterPrefix_V1))

#define MAX_BINARY_RECORD_ROUTER_TOS_PREFIX_SIZE \
    (sizeof(BinaryRecord_RouterTosPrefix_V1))

#define MAX_BINARY_RECORD_ROUTER_DESTONLY_SIZE \
    (sizeof(BinaryRecord_RouterDestOnly_V1))

#define MAX_BINARY_RECORD_ROUTER_SRCDEST_SIZE \
    (sizeof(BinaryRecord_RouterSrcDst_V1))

#define MAX_BINARY_RECORD_ROUTER_FULLFLOW_SIZE \
    (sizeof(BinaryRecord_RouterFullFlow_V1))

#define MAX_BINARY_RECORD_MPLSVPN_USAGE_SIZE \
    (sizeof(BinaryRecord_MPLSVPNUsage_V1))
#endif /* __NFC_DATAFILE_H__ */

```

**Note**

The **MPLSVPNUsage** aggregation scheme is not implemented in the FlowCollector. It is an aggregation scheme used for usage-based billing and implemented by the VPN Flow Collector of the Cisco Networking Services - Performance Engine (CNS-PE). You can access the CNS-PE documentation at: http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cnspe/1_0/index.htm.