

```

module fibrechannel {
  namespace "urn:t11:params:xml:ns:yang:fibrechannel";
  prefix fc;

  import ietf-interfaces {
    prefix if;
  }

  import iana-if-type {
    prefix ianaift;
  }

  import ietf-yang-types {
    prefix yang;
  }

  import fibrechannel-yang-types {
    prefix fc-types;
  }

  organization "T11 Technical Committee";

  contact
    "TC Web: <http://tools.ietf.org/wg/netmod/>
    WG List: <mailto:netmod@ietf.org>

    TC Chair: Steve Wilson
      <mailto:swilson@brocade.com>

    T11.3 Chair: Craig W. Carlson
      <mailto:craig.carlson@qlogic.com>

    FC-GS-8 Chair: Louis Ricci
      <mailto:louis.ricci@dell.com>

    FC-GS-7 Editor: Jason Rusch
      <mailto:jason.rusch@viavisolutions.com>";

  description
    "This module defines the YANG module 'fibrechannel', which augments the
    'interface' and 'interface-state' lists defined in the
    'ietf-interfaces' module [RFC7223] with Fibre Channel data nodes, and
    also adds Fibre Channel state data.

    Copyright © 2017 by Information Technology Industry Council (ITI)
    All rights reserved.";

  revision 2017-03-24 {
    description
      "Initial revision of YANG model for T11 Fibre Channel interfaces";
    reference
      "T11-3-2016, unless explicitly dated in the text.";
  }

```

```

/*

```

The following table provides specific implementation guidelines for applying the resources defined in the Interfaces Module to Fibre Channel interfaces. For those objects not listed here, refer to their generic definitions in [RFC7223]. (RFC 7223 takes precedence over these guidelines in the event of any conflict.)

Speed	For 1Gbs, this will be 1,000,000,000; for 2Gbs, it will be 2,000,000,000. If auto-negotiation is implemented and enabled on an interface, and the interface has not yet negotiated an operational speed, this object SHOULD reflect the maximum speed supported by the interface.
-------	---

phys-address	The interface's 24-bit Fibre Channel Address Identifier, or the zero-length string if no Address Identifier has been assigned to the interface.
--------------	---

admin-status Write access is not required, and support for 'testing' is not required.

oper-status Support for 'testing' is not required. The value 'dormant' has no meaning for Fibre Channel interfaces.

in-octets The number of octets of information contained in received frames between the Start-of-Frame and End-of-Frame delimiters (excluding the delimiters).

in-unicast-pkts The number of unicast frames received, i.e., the number of Start-of-Frame delimiters received for unicast frames.

in-errors The sum for this interface of  
 fcmPortLossOfSynchs  
 fcmPortLossOfSignals  
 fcmPortPrimSeqProtocolErrors  
 fcmPortInvalidTxWords  
 fcmPortInvalidCRCs  
 fcmPortAddressErrors  
 fcmPortDelimiterErrors  
 fcmPortTruncatedFrames  
 fcmPortEncodingDisparityErrors  
 plus any errors in fcmPortOtherErrors that were input errors.

out-octets The number of octets of information contained in transmitted frames between the Start-of-Frame and End-of-Frame delimiters (excluding the delimiters).

out-unicast-pkts The number of frames transmitted, i.e., the number of start-of-frame delimiters transmitted for unicast frames.

out-errors This is the number of errors in fcmPortOtherErrors that were output errors.

in-multicast-pkts These counters are not incremented  
 in-broadcast-pkts (unless a proprietary mechanism for  
 out-multicast-pkts multicast/broadcast is supported).  
 out-broadcast-pkts

\*/

```

/*
 * Configuration data nodes
 */
augment "/if:interfaces/if:interface" {
  when "if:type = 'ianaift:fibreChannel'" {
    description "Applies to all fibre channel interfaces";
  }
  description
    "Augment interface model with fibre channel specific configuration nodes";
}

container fibrechannel {
  description
    "Contains all fibre channel interface related configuration";
  leaf port-type {
    type fc-types:port-type-type;
    description "The specific mode currently enabled for the port.";
  }
  leaf-list supported-port-speeds {
    type identityref {

```

```
    base fc-types:fc-if-speed;
  }
  description "";
}
leaf speed {
  type identityref {
    base fc-types:fc-if-speed;
  }
  description
    "For PHY types that may operate at various speeds,
    this leaf allows the interface to be forced to
    operate at a particular speed. Without any explicit
    configuration, fibre channel interfaces run at the
    maximum speed that they are capable of operating
    at";
}
leaf port-name {
  type string {
    length "128"; /* what is the length */
    /* what are the character restrictions to define a pattern?*/
  }
  description
    "User friendly name.";
}
leaf class-of-service-capability {
  type fc-types:class-of-service-type;
  description
    "The classes of service capability of this port.";
}
leaf class-of-service-operational {
  type fc-types:class-of-service-type;
  description
    "The classes of service that are currently operational on
    this port. For an FL_Port, this is the union of the classes
    being supported across all attached NL_Ports.";
}
leaf transmitter-type {
  type enumeration {
    enum unknown {
      value 1;
      description
        "Unknown";
    }
    enum other {
      value 2;
      description
        "Other";
    }
    enum shortwave-850nm {
      value 3;
      description
        "Shortwave 850 nm";
    }
    enum shortwave-1550nm {
      value 4;
      description
        "Shortwave 1550 nm";
    }
    enum shortwave-1310nm {
      value 5;
      description
        "Shortwave 1310 nm";
    }
    enum electrical {
      value 6;
      description
        "Electrical";
    }
  }
  description
    "The technology of the port transceiver.";
  reference
    "FC-GS-3, section 6.1.2.2.3";
}
leaf connector-type {
  type enumeration {
```

```

enum unknown {
  value 1;
  description
    "Unknown";
}
enum other {
  value 2;
  description
    "Unknown";
}
enum gbic {
  value 3;
  description
    "Unknown";
}
enum embedded {
  value 4;
  description
    "Unknown";
}
enum glm {
  value 5;
  description
    "Unknown";
}
enum gbic-serial-id {
  value 6;
  description
    "Unknown";
}
enum gbic-no-serial-id {
  value 7;
  description
    "Unknown";
}
enum serial-id {
  value 8;
  description
    "Unknown";
}
enum no-serial-id {
  value 9;
  description
    "Unknown";
}
}
description
  "The module type of the port connector. This object refers
  to the hardware implementation of the port. It will be
  'embedded' if the hardware equivalent to Gigabit interface
  card (GBIC) is part of the line card and is unremovable. It
  will be 'glm' if it's a gigabit link module (GLM). It will
  be 'gbicSerialId' if the GBIC serial id can be read, else it
  will be 'gbicNoSerialId'. It will be 'sfpSerialId' if the
  small form factor (SFP) pluggable GBICs serial id can be
  read, else it will be 'sfpNoSerialId'.";
reference
  "FC-GS-3, section 6.1.2.2.4";
}
leaf serial-number {
  type string;
  description
    "The serial number associated with the port (e.g., for a
    GBIC). If not applicable, the object's value is a zero-
    length string.";
  reference
    "FC-GS-3, section 6.1.2.2.4";
}
leaf physical-port-number {
  type uint16;
  description
    "This is the port's 'Physical Port Number' as defined by
    GS-3.";
  reference
    "FC-GS-3, section 6.1.2.2.5";
}

```

```

leaf capable-protocol {
  type enumeration{
    enum unknown {
      value 0;
      description "";
    }
    enum loop {
      value 1;
      description "";
    }
    enum fabric {
      value 2;
      description "";
    }
    enum scsi {
      value 3;
      description "";
    }
    enum tcpip {
      value 4;
      description "";
    }
    enum vi {
      value 5;
      description "";
    }
    enum ficon {
      value 6;
      description "";
    }
  }
  description
  "A bit mask specifying the higher level protocols that are
  capable of running over this port. Note that for generic
  Fx_Ports, E_Ports, and B_Ports, this object will indicate
  all protocols.";
}
leaf operational-protocol {
  type enumeration{
    enum unknown {
      value 0;
      description "";
    }
    enum loop {
      value 1;
      description "";
    }
    enum fabric {
      value 2;
      description "";
    }
    enum scsi {
      value 3;
      description "";
    }
    enum tcpip {
      value 4;
      description "";
    }
    enum vi {
      value 5;
      description "";
    }
    enum ficon {
      value 6;
      description "";
    }
  }
  description
  "A bit mask specifying the higher level protocols that are
  currently operational on this port. For Fx_Ports, E_Ports,
  and B_Ports, this object will typically have the value
  'unknown'.";
}
}
}
}

```

```

/*
 * Operational State.
 */
augment "/if:interfaces-state/if:interface" {
  when "if:type = 'ianaift:fibreChannel'" {
    description "Applies to all fibre channel interfaces";
  }
  description
    "Augment interface model with fibre channel specific configuration nodes";

  container fibrechannel {
    description
      "Contains all fibre channel interface related configuration";
    leaf fcid {
      type fc-types:fcid-type;
      description "The Fibre Channel ID (FCID) of the port.";
    }
    leaf port-type {
      type fc-types:port-type-type;
      description "The specific mode currently enabled for the port.";
    }
    leaf speed {
      type identityref {
        base fc-types:fc-if-speed;
      }
      description
        "For PHY types that may operate at various speeds,
        this leaf allows the interface to be forced to
        operate at a particular speed. Without any explicit
        configuration, fibre channel interfaces run at the
        maximum speed that they are capable of operating
        at";
    }
    leaf-list supported-port-speeds {
      type identityref {
        base fc-types:fc-if-speed;
      }
      description "";
    }
    leaf reset {
      type boolean;
      description "Reset statistical counters.";
    }
    list neighbor {
      key "wwn";
      description
        "A list of wwn's ";
      leaf wwn {
        type fc-types:wwn-type;
        description "The Fibre Channel WWN of the neighbor port.";
      }
    }
  }
}
}
}
/*

```

```

fcmPortBBCreditZeros      Counter64,
fcmPortFullInputBuffers   Counter64,
fcmPortClass2RxFrames     Counter64,
fcmPortClass2RxOctets     Counter64,
fcmPortClass2TxFrames     Counter64,
fcmPortClass2TxOctets     Counter64,
fcmPortClass2Discards     Counter64,
fcmPortClass2RxFbsyFrames Counter64,
fcmPortClass2RxPbsyFrames Counter64,
fcmPortClass2RxFrjtFrames Counter64,
fcmPortClass2RxPrjtFrames Counter64,
fcmPortClass2TxFbsyFrames Counter64,
fcmPortClass2TxPbsyFrames Counter64,
fcmPortClass2TxFrjtFrames Counter64,
fcmPortClass2TxPrjtFrames Counter64,
fcmPortClass3RxFrames     Counter64,
fcmPortClass3RxOctets     Counter64,

```

```

fcmPortClass3TxFrames    Counter64,
fcmPortClass3TxOctets    Counter64,
fcmPortClass3Discards    Counter64,
fcmPortClassFRxFrames    Counter32,
fcmPortClassFRxOctets    Counter32,
fcmPortClassFTxFrames    Counter32,
fcmPortClassFTxOctets    Counter32,
fcmPortClassFDiscards    Counter32

```

## Errors

```

fcmPortRxLinkResets      Counter32,
fcmPortTxLinkResets      Counter32,
fcmPortLinkResets        Counter32,
fcmPortRxOfflineSequences Counter32,
fcmPortTxOfflineSequences Counter32,
fcmPortLinkFailures      Counter32,
fcmPortLossofSynchs      Counter32,
fcmPortLossofSignals     Counter32,
fcmPortPrimSeqProtocolErrors Counter32,
fcmPortInvalidTxWords    Counter32,
fcmPortInvalidCRCs       Counter32,
fcmPortInvalidOrderedSets Counter32,
fcmPortFrameTooLongs     Counter32,
fcmPortTruncatedFrames   Counter32,
fcmPortAddressErrors     Counter32,
fcmPortDelimiterErrors   Counter32,
fcmPortEncodingDisparityErrors Counter32,
fcmPortOtherErrors       Counter32

```

\*/

```

augment "/if: interfaces-state/if: interface/if: statistics" {
  when "if: type = 'ianaif: fibreChannel'" {
    description "Applies to all IEEE Std 802.3 Ethernet interfaces";
  }
  description
    "Augments 'statistics' container in ietf-interfaces/interfaces-statistics model.";
  container fibrechannel {
    description
      "";
    leaf sampling-interval {
      type uint16;
      description
        "Sampling interval in seconds.";
    }
    leaf time-generated {
      type yang:date-and-time;
      description
        "The time at which the statistics were queried";
    }
    leaf bb-credit-zeros {
      type yang:counter32;
      description
        "The number of transitions in/out of the buffer-to-buffer
        credit zero state. The other side is not providing any
        credit.";
    }
    leaf full-input-buffers {
      type yang:counter32;
      description
        "The number of occurrences when all input buffers of a port
        were full and outbound buffer-to-buffer credit transitioned
        to zero, i.e., there became no credit to provide to other
        side.";
    }
    leaf class2-in-frames {
      type yang:counter32;
      description
        "The number of Class 2 frames received at this port.";
    }
    leaf class2-in-octets {
      type yang:counter32;
      description
        "The number of octets contained in Class 2 frames received
        at this port.";
    }
    leaf class2-out-frames {
      type yang:counter32;

```

```
description
  "The number of Class 2 frames transmitted out of this port.";
}
leaf class2-out-octets {
  type yang:counter32;
  description
    "The number of octets contained in Class 2 frames transmitted
    out of this port.";
}
leaf class2-discards {
  type yang:counter32;
  description
    "The number of Class 2 frames that were discarded upon
    reception at this port.";
}
leaf class2-in-fbsy-frames {
  type yang:counter32;
  description
    "The number of times that F_BSY was returned to this port as
    a result of a Class 2 frame that could not be delivered to
    the other end of the link. This can occur when either the
    fabric or the destination port is temporarily busy. Note
    that this counter will never increment for an F_Port.";
}
leaf class2-in-pbsy-frames {
  type yang:counter32;
  description
    "The number of times that P_BSY was returned to this port as
    a result of a Class 2 frame that could not be delivered to
    the other end of the link. This can occur when the
    destination port is temporarily busy.";
}
leaf class2-in-frjt-frames {
  type yang:counter32;
  description
    "The number of times that F_RJT was returned to this port as
    a result of a Class 2 frame that was rejected by the fabric.
    Note that this counter will never increment for an F_Port.";
}
leaf class2-in-prjt-frames {
  type yang:counter32;
  description
    "The number of times that P_RJT was returned to this port as
    a result of a Class 2 frame that was rejected at the
    destination N_Port.";
}
leaf class2-out-fbsy-frames {
  type yang:counter32;
  description
    "The number of times that F_BSY was generated by this port
    as a result of a Class 2 frame that could not be delivered
    because either the Fabric or the destination port was
    temporarily busy. Note that this counter will never
    increment for an N_Port.";
}
leaf class2-out-pbsy-frames {
  type yang:counter32;
  description
    "The number of times that P_BSY was generated by this port
    as a result of a Class 2 frame that could not be delivered
    because the destination port was temporarily busy. Note
    that this counter will never increment for an F_Port.";
}
leaf class2-out-frjt-frames {
  type yang:counter32;
  description
    "The number of times that F_RJT was generated by this port
    as a result of a Class 2 frame being rejected by the fabric.
    Note that this counter will never increment for an N_Port.";
}
leaf class2-out-prjt-frames {
  type yang:counter32;
  description
    "The number of times that P_RJT was generated by this port
    as a result of a Class 2 frame being rejected at the
    destination N_Port. Note that this counter will never
```



```
    increment for an F_Port.";
}
leaf class3-in-frames {
    type yang:counter32;
    description
        "The number of Class 3 frames received at this port.";
}
leaf class3-in-octets {
    type yang:counter32;
    description
        "The number of octets contained in Class 3 frames received
        at this port.";
}
leaf class3-out-frames {
    type yang:counter32;
    description
        "The number of Class 3 frames transmitted out of this port.";
}
leaf class3-out-octets {
    type yang:counter32;
    description
        "The number of octets contained in Class 3 frames
        transmitted out of this port.";
}
leaf class3-discards {
    type yang:counter32;
    description
        "The number of Class 3 frames that were discarded upon
        reception at this port.";
}

leaf in-link-resets {
    type yang:counter32;
    description
        "The number of Link Reset (LR) Primitive Sequences
        received.";
}
leaf out-link-resets {
    type yang:counter32;
    description
        "The number of Link Reset (LR) Primitive Sequences
        transmitted.";
}
leaf link-resets {
    type yang:counter32;
    description
        "The number of times the reset link protocol was initiated
        on this port. This includes the number of Loop
        Initialization Primitive (LIP) events on an arbitrated loop
        port.";
}
leaf in-offline-sequences {
    type yang:counter32;
    description
        "The number of times the reset link protocol was initiated
        on this port. This includes the number of Loop
        Initialization Primitive (LIP) events on an arbitrated loop
        port.";
}
leaf out-offline-sequences {
    type yang:counter32;
    description
        "The number of Offline (OLS) Primitive Sequences transmitted
        by this port.";
}
leaf link-failures {
    type yang:counter32;
    description
        "The number of link failures. This count is part of FC-PH's
        Link Error Status Block (LESB).";
    reference
        "FC-PH, rev 4.3, 1 June 1994, section 29.8 [FC-PH].";
}
leaf loss-of-synchs {
```

```
type yang:counter32;
description
  "The number of instances of synchronization loss detected at
  this port. This count is part of FC-PH's Link Error Status
  Block (LESB).";
reference
  "FC-PH, rev 4.3, 1 June 1994, section 29.8.";
}
leaf loss-of-signal {
  type yang:counter32;
  description
    "The number of instances of signal loss detected at this
    port. This count is part of FC-PH's Link Error Status Block
    (LESB).";
  reference
    "FC-PH, rev 4.3, 1 June 1994, section 29.8.";
}
leaf prim-seq-protocol-errors {
  type yang:counter32;
  description
    "The number of primitive sequence protocol errors detected
    at this port. This count is part of FC-PH's Link Error
    Status Block (LESB).";
  reference
    "FC-PH, rev 4.3, 1 June 1994, section 29.8.";
}
leaf invalid-out-words {
  type yang:counter32;
  description
    "The number of invalid transmission words received at this
    port. This count is part of FC-PH's Link Error Status Block
    (LESB).";
  reference
    "FC-PH, rev 4.3, 1 June 1994, section 29.8.";
}
leaf invalid-cracs {
  type yang:counter32;
  description
    "The number of frames received with an invalid CRC. This
    count is part of FC-PH's Link Error Status Block (LESB).";
  reference
    "FC-PH, rev 4.3, 1 June 1994, section 29.8.";
}
leaf invalid-ordered-sets {
  type yang:counter32;
  description
    "The number of invalid ordered sets received at this port.";
}
leaf truncated-frames {
  type yang:counter32;
  description
    "The number of frames received at this port for which the
    frame length was greater than what was agreed to in
    FLOGI/PLOGI. This could be caused by losing the end of
    frame delimiter.";
}
leaf address-errors {
  type yang:counter32;
  description
    "The number of frames received with unknown addressing; for
    example, an unknown SID or DID.";
}
leaf delimiter-errors {
  type yang:counter32;
  description
    "The number of invalid frame delimiters received at this
    port. An example is a frame with a class 2 start and a
    class 3 at the end.";
}
leaf encoding-disparity-errors {
  type yang:counter32;
  description
    "The number of invalid frame delimiters received at this
    port. An example is a frame with a class 2 start and a
    class 3 at the end.";
}
```

```
leaf other-errors {  
  type yang:counter32;  
  description  
    "The number of errors that were detected on this port but  
    not counted by any other error counter in this row."  
}  
}  
}
```