WANCommonInterfaceConfig:1 Service Template Version 1.01

For UPnP™ Version 1.0
Status: Standardized DCP
Date: November 12, 2001

Authors	Company
Ulhas Warrier, Prakash Iyer	Intel Corporation

Contents

List of Tables

1. Overview and Scope

This service definition is compliant with the UPnP Device Architecture version *1.0*.

This service-type models physical layer properties of a WAN interface on an Internet Gateway used for Internet access.

The service is REQUIRED and is specified in
urn:schemas-upnp-org:device:*WANDevice*
one or more instances of which are specified under the root device
urn:schemas-upnp-org:device:*InternetGatewayDevice*

1.1. Change Log

Version 0.6 of this document replaced ***WANInternetAccess:0.5***.
Changes from ***WANInternetAccess:0.5***

- Renamed service to *WANCommonInterfaceConfig*
- Removed `LinkStatus` state variable (it is now part of the Interface Configuration service corresponding to a specific WAN access type e.g,, *WANPOTSInterfaceConfig*)
- Changed error return codes in conformance with latest device template.
- Added ‘Ethernet’ to list of allowed values for WAN access type.

Changes from ***WANCommonInterfaceConfig:0.6***

- Added ‘Get’ actions per Technical Committee recommendation to not use QueryStateVariable for reading state variables.

Changes from ***WANCommonInterfaceConfig:0.7*** (as per WC meeting on 9/6/00)

- Change ADSL to DSL in SST
- Removed “(or PPP)” from EnabledForInternet description
- Reduce number of GET functions defining a new aggregate function – GetWANAccessProperties
- Removed <retval/> and empty defaultvalue tags from XML description doc – not needed.
- Removed values “satellite” and “wireless” from AllowedValueList for WANAccessType
- Changed Boolean values from true/false to 0/1
- Moved LinkStatus from WANXYZInterfaceConfig

Changes from ***WANCommonInterfaceConfig:0.8*** (as per WC meeting on 10/17/00)

- Changes to reflect required and optional SST variables per discussion at the F2F meeting
- Added new SST variables TotalBytesSent, TotalBytesReceived, TotalPacketsSent, TotalPacketsReceived
- Split Get actions into required and optional to reflect changes to SST variables
- Changed ActiveClients to an array of 2-tuples
- Updates to Theory of operation section.

Changes from ***WANCommonInterfaceConfig:0.9***

- Added Initializing to AllowedValueList for PhysicalLinkStatus

Changes from ***WANCommonInterfaceConfig:0.91***

- Removed “urn” from ActiveConnectionDeviceContainer value
- Removed white spaces from XML section

Changes from ***WANCommonInterfaceConfig:0.92***

- Changed document status to template design complete
- Clarification to text in many sections of the draft
- Added prefix “New” to formal parameters for actions to distinguish them from related state variables
- Added action GetEnabledForInternet
- Removed ‘retval’ tags from the XML specification.

Changes from ***WANCommonInterfaceConfig:0.93***

- Updated to service template v1.01

- Verified against TDC checklist v1.01
- Down rev'ed document version to comply with TDC requirements
- Modified NewActiveConnectionDeviceContainer to NewActiveConnDeviceContainer to comply with 32-character limit.
- Added a note to description of MaximumActiveConnections.
- Deleted Other from the allowedValueList for WANAccessType
- ReplacedGetCommonConnectionProperties with 6 individual Get actions.

Changes from ***WANCommonInterfaceConfig:0.8***

- Removed default values and updated XML section accordingly
- Updated allowedValueList tables to clarify Required versus Optional values
- Deleted Vendor Defined rows from allowedValueList tables

Changes from ***WANCommonInterfaceConfig:0.81***

- Added XML comment tags to comments text in XML template
- Renamed ActiveConnectionIndex to A_ActiveConnectionIndex and updated SST and event tables.

Changes from ***WANCommonInterfaceConfig:0.82***

- Updated semantic tests section
- Deleted A_ActiveConnectionIndex and updated related text and XML sections
- Deleted allowedvaluerange for boolean variables in XML template

Changes from ***WANCommonInterfaceConfig:0.9***

- Deleted mention of 'Other' as an allowed value for WANAccessType in section 2.2.1
- Changed variable name string 'NumberofActiveConnections' to 'NumberOfActiveConnections' for naming convention and consistency with xml section.
- Deleted empty <maximum> tag for MaximumActiveConnections from XML section

Changes from ***WANCommonInterfaceConfig:0.99***

- Version updated to reflect 45-day review completion. No other changes to this draft.

Changes from ***WANCommonInterfaceConfig:0.991***

- Copyright messages and document status updated.

2. Service Modeling Definitions

2.1. ServiceType

The following service type identifies a service that is compliant with this template:

urn:schemas-upnp-org:service:WANCommonInterfaceConfig:1.

2.2. State Variables

Table 1: State Variables

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value [2]	Default Value [2]	Eng. Units
`WANAccessType`	R	string	See table 1.1 below	Empty string	N/A
`Layer1UpstreamMaxBitRate`	R	ui4	Range of values is WAN device, technology and/or deployment scenario specific	Undefined	bitspersecond
`Layer1DownstreamMaxBitRate`	R	ui4	Range of values is WAN device, technology and/or deployment scenario specific	Undefined	bitspersecond
`PhysicalLinkStatus`	R	string	See table 1.2 below	Not specified	N/A
`WANAccessProvider`	O	string	No specific range of values defined	Empty string	N/A
`MaximumActiveConnections`	O	ui2	A fixed number, product dependent.	A fixed number, product dependent	N/A
`NumberOfActiveConnections`	O	ui2	Undefined	Not specified	N/A

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value [2]	Default Value [2]	Eng. Units
ActiveConnectionDeviceContainer	O	string	Undefined	Empty string	N/A
ActiveConnectionServiceID	O	string	Undefined	Empty string	N/A
TotalBytesSent	O	ui4	Undefined	Not specified	N/A
TotalBytesReceived	O	ui4	Undefined	Not specified	N/A
TotalPacketsSent	O	ui4	Undefined	Not specified	N/A
TotalPacketsReceived	O	ui4	Undefined	Not specified	N/A
EnabledForInternet	O	boolean	0,1	Not specified	N/A
Non-standard state variables implemented by an UPnP vendor go here.	*X*	*TBD*	*TBD*	*TBD*	*TBD*

[1] R = Required, O = Optional, X = Non-standard.

[2] Values listed in this column are required. To specify standard optional values or to delegate assignment of values to the vendor, you must reference a specific instance of an appropriate table below.

NOTE: Default values are not specified in the DCP. A vendor may however choose to provide default values for SST variables where appropriate.

Table 1.1: allowedValueList for WANAccessType

Value	Req. or Opt. [3]	Description
DSL	R	*Only applicable for Digital Subscriber Link (DSL) modems*
POTS	R	*Only applicable for Plain Old Telephone Service (POTS) modems*
Cable	R	*Only applicable for cable modems*
Ethernet	R	*Only applicable for external Ethernet-attached modems*

[3] *WANAccessType must be set to only one of the values listed in the table, as appropriate for the hardware device.*

Table 1.2: allowedValueList for PhysicalLinkStatus

Value	Req. or Opt. [3]
Up	R
Down	R
Initializing	O
Unavailable	O

[3]

2.2.1. WANAccessType

This variable specifies the type of WAN access (modem) between the residential network and the Internet Service Provider (ISP). *Ethernet* refers to an Ethernet-attached external modem. Other values are self-explanatory.

2.2.2. Layer1UpstreamMaxBitRate

This variable specifies the maximum upstream (from the ***WANDevice*** to ISP) theoretical bit rate (in bits per second) for the WAN device. For example, 33600 for a POTS V.90 modem.

2.2.3. Layer1DownstreamMaxBitRate

This variable specifies the maximum downstream (from the ISP to ***WANDevice***) theoretical bit rate (in bits per second) for the WAN device. For example, 56000 for a POTS V.90 modem.

2.2.4. PhysicalLinkStatus

This variable indicates the state of the physical connection (link) from ***WANDevice*** to a connected entity (could be ISP CO for an integrated modem and Ethernet link status for an external Ethernet-connected modem).

2.2.5. WANAccessProvider

This is a descriptive string name of the Service Provider providing link connectivity on the WAN interface. This provider MAY or MAY NOT also be the Internet Service Provider. For example, a customer MAY have DSL service from a phone company and Internet Access Service from another. In this case, the **former** is identified by **this** variable. The format of the text is implementation dependent. An implementation may provide information such as customer support phone numbers so a user can get help when needed.

2.2.6. MaximumActiveConnections

This variable indicates the maximum number of active connections the gateway can simultaneously support. This may be different from the number of instances of ***WAN*Connection*** service that is initialized by the gateway. For example, the description document may provide 10

static instances of ***WANPPPConnection***. But if the value of this variable is 5, no more than 5 connections can be simultaneously active on this ***WANDevice***.
Note that the value of this variable may be dependent on the software and hardware capabilities of a specific WAN interface (modeled by a ***WANDevice***) as well as a function of the type of connections supported. For example, a certain modem may support up to 5 simultaneous bridged connections but only 3 simultaneous PPTP connections. This value cannot be 0.

2.2.7. NumberOfActiveConnections

Number of ***WAN*Connection*** service instances currently active on a WAN interface.

2.2.8. ActiveConnectionDeviceContainer & ActiveConnectionServiceID

Active Connections are represented as an array of entries – each entry consists of the 2 values represented by the following state variables -

ActiveConnectionDeviceContainer – Identifies the ***WANConnectionDevice*** container for the active connection service instance. Has the form:
uuid:*device-UUID*:*WANConnectionDevice*:*v*
ActiveConnectionServiceID - Service ID of the active connection service. Has the form:
urn:**upnp-org**:**serviceId**:*serviceID*.

The variable NumberOfActiveConnections indicates the total number of 'simultaneously active connections' (elements in the array) on this ***WANDevice***. To retrieve all array entries a client should iterate GetActiveConnection calls incrementing the array index from 0 to [NumberOfActiveConnections-1].

2.2.9. TotalBytesSent

This variable represents the cumulative counter for total number of bytes sent upstream across all connection service instances on ***WANDevice***. The count rolls over to 0 after it reaching the maximum value (2^32) –1.

2.2.10.TotalBytesReceived

This variable represents the cumulative counter for total number of bytes received downstream across all connection service instances on ***WANDevice***. The count rolls over to 0 after it reaching the maximum value (2^32) –1.

2.2.11.TotalPacketsSent

This variable represents the cumulative counter for total number of IP or PPP packets sent upstream across all connection service instances on ***WANDevice***. The count rolls over to 0 after it reaching the maximum value (2^32) –1.

2.2.12.TotalPacketsReceived

This variable represents the cumulative counter for total number of IP or PPP packets received downstream across all connection service instances on ***WANDevice***. The count rolls over to 0 after it reaching the maximum value (2^32) –1.

2.2.13.EnabledForInternet

This variable can be used to enable or disable access to and from the Internet, across all connection instances. It can be set to **0** to turn off access to the Internet. In the case of Always-On, Always-Connected (AOAC) technologies such as DSL or cable modems, IP or PPP sessions will be disabled at the ***WANDevice***.

2.2.14.Relationships Between State Variables

If EnabledForInternet is set to 0, NumberOfActiveConnections should be ignored. NumberOfActiveConnections cannot exceed MaximumActiveConnections.

NumberOfActiveConnections, ActiveConnectionDeviceContainer and ActiveConnectionServiceID are related and a vendor that chooses to implement this feature MUST implement all 3 of these variables.

2.3. Eventing and Moderation

Table 2: Event Moderation

Variable Name	Evented	Moderated Event	Max Event Rate[1]	Logical Combination	Min Delta per Event[2]
WANAccessType	No	No	N/A	N/A	N/A
Layer1UpstreamMaxBitRate	No	No	N/A	N/A	N/A
Layer1DownstreamMaxBitRate	No	No	N/A	N/A	N/A
PhysicalLinkStatus	Yes	No	N/A	N/A	N/A
WANAccessProvider	No	No	N/A	N/A	N/A
MaximumActiveConnections	No	No	N/A	N/A	N/A
NumberOfActiveConnections	Yes	No	N/A	N/A	N/A
ActiveConnectionDeviceContainer	No	No	N/A	N/A	N/A
ActiveConnectionServiceID	No	No	N/A	N/A	N/A
TotalBytesSent	No	No	N/A	N/A	N/A
TotalBytesReceived	No	No	N/A	N/A	N/A

Variable Name	Evented	Moderated Event	Max Event Rate[1]	Logical Combination	Min Delta per Event[2]
TotalPacketsSent	No	No	N/A	N/A	N/A
TotalPacketsReceived	No	No	N/A	N/A	N/A
EnabledForInternet	Yes	No	N/A	N/A	N/A
Non-standard state variables implemented by an UPnP vendor go here.	*TBD*	*TBD*	*TBD*	*TBD*	*TBD*

[1] Determined by N, where Rate = (Event)/(N secs).
[2] (N) * (allowedValueRange Step).

2.3.1. Event Model

The eventing model is simple and is not moderated. Clients registered for events on this service can use updates on PhysicalLinkStatus, NumberOfActiveConnections and EnabledForInternet to provide user feedback and manage connections from local applications.

2.4. Actions

Immediately following this table is detailed information about these actions, including short descriptions of the actions, the effects of the actions on state variables, and error codes defined by the actions.

Table 3: Actions

Name	Req. or Opt. [1]
SetEnabledForInternet	O
GetEnabledForInternet	O
GetCommonLinkProperties	R
GetWANAccessProvider	O
GetMaximumActiveConnections	O
GetTotalBytesSent	O
GetTotalBytesReceived	O
GetTotalPacketsSent	O
GetTotalPacketsReceived	O
GetActiveConnection	O
Non-standard actions implemented by an UPnP vendor go here.	X

[1] R = Required, O = Optional, X = Non-standard.

2.4.1. SetEnabledForInternet

This action enables or disables access to the Internet on the ***WANDevice***.

2.4.1.1. Arguments

Table 4: Arguments for `SetEnabledForInternet`

Argument	Direction	relatedStateVariable
`NewEnabledForInternet`	*IN*	`EnabledForInternet`

2.4.1.2. Dependency on State (if any)

None.

2.4.1.3. Effect on State (if any)

Setting this variable to 0 turns off access to the Internet on this ***WANDevice***. Actions in other services may become "Don't Care" operations in this state.

2.4.1.4. Errors

errorCode	errorDescription	Description
401	Invalid Action	No action by that name at this service.
402	Invalid Args	One of following: not enough IN arguments, too many IN arguments, no IN argument by that name, one or more IN arguments are of the wrong data type. See also the UPnP Device Architecture.
501	Action Failed	May be returned in current state if service prevents invoking of that action.

2.4.2. GetEnabledForInternet

This action retrieves the value of `EnabledForInternet`.

2.4.2.1. Arguments

Table 5: Arguments for `GetEnabledForInternet`

Argument	Direction	relatedStateVariable
`NewEnabledForInternet`	*OUT*	`EnabledForInternet`

2.4.2.2. Dependency on State (if any)

2.4.2.3. Effect on State

None.

2.4.2.4. Errors

errorCode	errorDescription	Description
401	Invalid Action	No action by that name at this service.

402	Invalid Args	One of following: not enough IN arguments, too many IN arguments, no IN argument by that name, one or more IN arguments are of the wrong data type. See also the UPnP Device Architecture.
501	Action Failed	May be returned in current state if service prevents invoking of that action.

2.4.3. GetCommonLinkProperties

This action retrieves physical link properties of the WAN interface (***WANDevice***).

2.4.3.1. Arguments

Table 6: Arguments for `GetCommonLinkProperties`

Argument	Direction	relatedStateVariable
`NewWANAccessType`	*OUT*	`WANAccessType`
`NewLayer1UpstreamMaxBitRate`	*OUT*	`Layer1UpstreamMaxBitRate`
`NewLayer1DownstreamMaxBitRate`	*OUT*	`Layer1DownstreamMaxBitRate`
`NewPhysicalLinkStatus`	*OUT*	`PhysicalLinkStatus`

2.4.3.2. Dependency on State (if any)

2.4.3.3. Effect on State

None.

2.4.3.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	One of following: not enough IN arguments, too many IN arguments, no IN argument by that name, one or more IN arguments are of the wrong data type. See also the UPnP Device Architecture.

2.4.4. GetWANAccessProvider

This action retrieves a string description of the link service provider, common to all connection instances on a ***WANDevice***.

2.4.4.1. Arguments

Table 7: Arguments for `GetWANAccessProvider`

Argument	Direction	relatedStateVariable
NewWANAccessProvider	OUT	WANAccessProvider

2.4.4.2. Dependency on State (if any)

2.4.4.3. Effect on State

None.

2.4.4.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	One of following: not enough IN arguments, too many IN arguments, no IN argument by that name, one or more IN arguments are of the wrong data type. See also the UPnP Device Architecture.

2.4.5. GetMaximumActiveConnections

This action retrieves the maximum number of active connections possible across all connection instances on a *WANDevice*.

2.4.5.1. Arguments

Table 8: Arguments for `GetMaximumActiveConnections`

Argument	Direction	relatedStateVariable
NewMaximumActiveConnections	OUT	MaximumActiveConnections

2.4.5.2. Dependency on State (if any)

2.4.5.3. Effect on State

None.

2.4.5.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	One of following: not enough IN arguments, too many IN arguments, no IN argument by that name, one or more IN arguments are of the wrong data type. See also the UPnP Device Architecture.

2.4.6. GetTotalBytesSent

This action retrieves the cumulative count of bytes sent upstream across all connection instances on a *WANDevice*.

2.4.6.1. Arguments

Table 9: Arguments for `GetTotalBytesSent`

Argument	Direction	relatedStateVariable
`NewTotalBytesSent`	*OUT*	`TotalBytesSent`

2.4.6.2. Dependency on State (if any)

2.4.6.3. Effect on State

None.

2.4.6.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	One of following: not enough IN arguments, too many IN arguments, no IN argument by that name, one or more IN arguments are of the wrong data type. See also the UPnP Device Architecture.

2.4.7. GetTotalBytesReceived

This action retrieves the cumulative count of bytes received downstream across all connection instances on a ***WANDevice***.

2.4.7.1. Arguments

Table 10: Arguments for `GetTotalBytesReceived`

Argument	Direction	relatedStateVariable
`NewTotalBytesReceived`	*OUT*	`TotalBytesReceived`

2.4.7.2. Dependency on State (if any)

2.4.7.3. Effect on State

None.

2.4.7.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	One of following: not enough IN arguments, too many IN arguments, no IN argument by that name, one or more IN arguments are of the wrong data type. See also the UPnP Device Architecture.

2.4.8. GetTotalPacketsSent

This action retrieves the cumulative count of IP or PPP packets sent upstream across all connection instances on a ***WANDevice***.

2.4.8.1. Arguments

Table 11: Arguments for `GetTotalPacketsSent`

Argument	Direction	relatedStateVariable
`NewTotalPacketsSent`	*OUT*	`TotalPacketsSent`

2.4.8.2. Dependency on State (if any)

2.4.8.3. Effect on State

None.

2.4.8.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	One of following: not enough IN arguments, too many IN arguments, no IN argument by that name, one or more IN arguments are of the wrong data type. See also the UPnP Device Architecture.

2.4.9. GetTotalPacketsReceived

This action retrieves the cumulative count of IP or PPP packets received downstream across all connection instances on a ***WANDevice***.

2.4.9.1. Arguments

Table 12: Arguments for `GetTotalPacketsReceived`

Argument	Direction	relatedStateVariable
`NewTotalPacketsReceived`	*OUT*	`TotalPacketsReceived`

2.4.9.2. Dependency on State (if any)

2.4.9.3. Effect on State

None.

2.4.9.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	One of following: not enough IN arguments, too many IN arguments, no IN argument by that name, one or more IN arguments are of the wrong data type. See also the UPnP Device Architecture.

2.4.10.GetActiveConnection

Retrieves the service ID of a specific active connection. Active connections are represented as an array. If a client wishes to get all the elements of the array, it repeats this action looping on the array index from 0 to (`NumberOfActiveConnections` –1) OR till an error is returned. Note that the client should reevaluate its loop if `NumberOfActiveConnections` is evented during the process of getting all entries.

2.4.10.1.Arguments

Table 13: Arguments for `GetActiveConnection`

Argument	Direction	relatedStateVariable
`NewActiveConnectionIndex`	*IN*	`NumberOfActiveConnections`
`NewActiveConnDeviceContainer` `NewActiveConnectionServiceID`	*OUT* *OUT*	`ActiveConnectionDeviceContainer` `ActiveConnectionServiceID`

2.4.10.2.Dependency on State (if any)

2.4.10.3.Effect on State

None.

2.4.10.4.Errors

errorCode	errorDescription	Description
402	Invalid Args	One of following: not enough IN arguments, too many IN arguments, no IN argument by that name, one or more IN arguments are of the wrong data type. See also the UPnP Device Architecture.
501	Action Failed	May be returned in current state if service prevents invoking of that action.
712	NullValueAtSpecifiedArrayIndex	The specified array index holds a NULL value

2.4.11.Non-Standard Actions Implemented by a UPnP Vendor

To facilitate certification, non-standard actions implemented by UPnP vendors should be included in this service template. The UPnP Device Architecture lists naming requirements for non-standard actions (see the section on Description).

2.4.12.Relationships Between Actions

If `SetEnabledForInternet` sets `EnabledForInternet` to 0, set actions attempting to initiate connections in other services may return errors.

2.4.13.Common Error Codes

The following table lists error codes common to actions for this service type. If an action results in multiple errors, the most specific error should be returned.

Table 14: Common Error Codes

errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture section on Control.
402	Invalid Args	See UPnP Device Architecture section on Control.
404	Invalid Var	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
600-699	TBD	Common action errors. Defined by UPnP Forum Technical Committee.
701-799		Common action errors defined by the UPnP Forum working committees.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.5. Theory of Operation

The ***WANCommonInterfaceConfig*** service MUST be implemented for each ***WANDevice***. The service models WAN interface properties common across all connection service instances.

`MaximumActiveConnections` is specific to the type of WAN interface. The relationship between the static number of instances of ***WAN*Connection*** service in a description document and the values of `MaximumActiveConnections` and `NumberOfActiveConnections` is best illustrated with an example.

Suppose a gateway supports a POTS modem on its WAN interface. To support the possibility of multiple connection instances (one for each user account with an ISP for example), a gateway vendor may specific a static number of ***WANPPPConnection*** instances. However, as only one connection instance can be active on the POTS modem at any time, `MaximumActiveConnections` will be set to 1. `ActiveConnectionServiceID` corresponding to the first entry (in the array of active connections) will designate the service ID of the ***WANPPPConnection*** that is currently active (in use).

3. XML Service Description

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action>
    <name>SetEnabledForInternet</name>
      <argumentList>
        <argument>
          <name>NewEnabledForInternet</name>
          <direction>in</direction>
         <relatedStateVariable>EnabledForInternet</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetEnabledForInternet</name>
      <argumentList>
        <argument>
          <name>NewEnabledForInternet</name>
          <direction>out</direction>
         <relatedStateVariable>EnabledForInternet</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetCommonLinkProperties</name>
      <argumentList>
        <argument>
          <name>NewWANAccessType</name>
          <direction>out</direction>
          <relatedStateVariable>WANAccessType</relatedStateVariable>
        </argument>
        <argument>
          <name>NewLayer1UpstreamMaxBitRate</name>
          <direction>out</direction>
   <relatedStateVariable>Layer1UpstreamMaxBitRate</relatedStateVariable>
        </argument>
        <argument>
          <name>NewLayer1DownstreamMaxBitRate</name>
          <direction>out</direction>
<relatedStateVariable>Layer1DownstreamMaxBitRate</relatedStateVariable>
        </argument>
        <argument>
          <name>NewPhysicalLinkStatus</name>
          <direction>out</direction>
         <relatedStateVariable>PhysicalLinkStatus</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetWANAccessProvider</name>
      <argumentList>
        <argument>
          <name>NewWANAccessProvider</name>
          <direction>out</direction>
          <relatedStateVariable>WANAccessProvider</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetMaximumActiveConnections</name>
      <argumentList>
        <argument>
          <name>NewMaximumActiveConnections</name>
          <direction>out</direction>
   <relatedStateVariable>MaximumActiveConnections</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetTotalBytesSent</name>
      <argumentList>
        <argument>
          <name>NewTotalBytesSent</name>
          <direction>out</direction>
          <relatedStateVariable>TotalBytesSent</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetTotalBytesReceived</name>
      <argumentList>
        <argument>
          <name>NewTotalBytesReceived</name>
          <direction>out</direction>
          <relatedStateVariable>TotalBytesReceived</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetTotalPacketsSent</name>
      <argumentList>
        <argument>
          <name>NewTotalPacketsSent</name>
          <direction>out</direction>
          <relatedStateVariable>TotalPacketsSent</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetTotalPacketsReceived</name>
      <argumentList>
        <argument>
          <name>NewTotalPacketsReceived</name>
          <direction>out</direction>
         <relatedStateVariable>TotalPacketsReceived</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetActiveConnection</name>
      <argumentList>
        <argument>
          <name>NewActiveConnectionIndex</name>
          <direction>in</direction>
<relatedStateVariable>NumberOfActiveConnections</relatedStateVariable>
        </argument>
        <argument>
          <name>NewActiveConnDeviceContainer</name>
          <direction>out</direction>
<relatedStateVariable>ActiveConnectionDeviceContainer</relatedStateVaria
ble>
        </argument>
        <argument>
          <name>NewActiveConnectionServiceID</name>
          <direction>out</direction>
<relatedStateVariable>ActiveConnectionServiceID</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <!-- Declarations for other actions added by UPnP vendor (if any) go
here -->
  </actionList>
  <serviceStateTable>
    <stateVariable sendEvents="no">
      <name>WANAccessType</name>
      <dataType>string</dataType>
      <allowedValueList>
        <allowedValue>DSL</allowedValue>
        <allowedValue>POTS</allowedValue>
        <allowedValue>Cable</allowedValue>
        <allowedValue>Ethernet</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>Layer1UpstreamMaxBitRate</name>
      <dataType>ui4</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>Layer1DownstreamMaxBitRate</name>
      <dataType>ui4</dataType>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>PhysicalLinkStatus</name>
      <dataType>string</dataType>
      <allowedValueList>
        <allowedValue>Up</allowedValue>
        <allowedValue>Down</allowedValue>
        <allowedValue>Initializing</allowedValue>
        <allowedValue>Unavailable</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>WANAccessProvider</name>
      <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>MaximumActiveConnections</name>
      <dataType>ui2</dataType>
      <allowedValueRange>
        <minimum>1</minimum>
        <step>1</step>
      </allowedValueRange>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>NumberOfActiveConnections</name>
      <dataType>ui2</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>ActiveConnectionDeviceContainer</name>
      <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>ActiveConnectionServiceID</name>
      <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>TotalBytesSent</name>
      <dataType>ui4</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>TotalBytesReceived</name>
      <dataType>ui4</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>TotalPacketsSent</name>
      <dataType>ui4</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>TotalPacketsReceived</name>
      <dataType>ui4</dataType>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>EnabledForInternet</name>
      <dataType>boolean</dataType>
    </stateVariable>
    <!-- Declarations for other state variables added by UPnP vendor (if
any) go here -->
  </serviceStateTable>
</scpd>
```

4. Test

SetEnabledForInternet / GetEnabledForInternet

Test Sequence 1: To test success path
Semantic class: 2
Pre-conditions:

- Follow sequence of actions outlined in the WANPPP or WANIPConnection service descriptions to ensure that `ConnectionStatus` is Disconnected.

NOTE: This test is only applicable if the vendor implements the feature.

SetEnabledForInternet Success = 200

In-Arg	Values	State Variables	Current State	Expected State
EnabledForInternet	1			
Out-Arg	Expected Value			
		Error Code (if any)	NA	NA

GetEnabledForInternet Success = 200

In-Arg	Values	State Variables	Current State	Expected State
Out-Arg	Expected Value			
EnabledForInternet	1	Error Code (if any)	NA	NA

This test should be repeated with the value set to 0.

Change History

Change Log for Version 1.0 (10-4-00)

- Revised the Title Page to call out V1.0 of the Service Template
- Changed to be consistent with Sample Designs released to the Technical Committee
- Service State Table: Variable Descriptions removed from the table and are listed in specific sections following the table.
- Actions: Reformatted the information contained in the Action Table:
  - Added overview entry point.
  - Added an Action Summary Table to specify Required or Optional
  - Added enumerated sections to specify each actions: Arguments, Effect on State, and Errors.