
WANPOTSLinkConfig:1 Service Template Version 1.01

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

This Standardized DCP has been adopted as a Standardized DCP by the Steering Committee of the UPnP™ Forum, pursuant to Section 2.1(c)(ii) of the UPnP™ Forum Membership Agreement. UPnP™ Forum Members have rights and licenses defined by Section 3 of the UPnP™ Forum Membership Agreement to use and reproduce the Standardized DCP in UPnP™ Compliant Devices. All such use is subject to all of the provisions of the UPnP™ Forum Membership Agreement.

THE UPNP™ FORUM TAKES NO POSITION AS TO WHETHER ANY INTELLECTUAL PROPERTY RIGHTS EXIST IN THE STANDARDIZED DCPS. THE STANDARDIZED DCPS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". THE UPNP™ FORUM MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE STANDARDIZED DCPS, INCLUDING BUT NOT LIMITED TO ALL IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OF REASONABLE CARE OR WORKMANLIKE EFFORT, OR RESULTS OR OF LACK OF NEGLIGENCE.

© 1999-2001 Contributing Members of the UPnP™ Forum. All Rights Reserved.

Authors	Company
Ulhas Warriar, Prakash Iyer	Intel Corporation

Contents

1. OVERVIEW AND SCOPE	4
1.1. CHANGE LOG	4
2. SERVICE MODELING DEFINITIONS	5
2.1. SERVICE TYPE	5
2.2. STATE VARIABLES	5
2.2.1. ISPPhoneNumber	7
2.2.2. ISPInfo	8
2.2.3. LinkType	8
2.2.4. NumberOfRetries	8
2.2.5. DelayBetweenRetries	8
2.2.6. Fclass	8
2.2.7. DataModulationSupported	8
2.2.8. DataProtocol	8
2.2.9. DataCompression	8
2.2.10. PlusVTRCommandSupported	8
2.2.11. Relationships Between State Variables	8
2.3. EVENTING AND MODERATION	8
2.3.1. Event Model	9
2.4. ACTIONS	9
2.4.1. SetISPInfo	10
2.4.2. SetCallRetryInfo	10
2.4.3. GetISPInfo	11
2.4.4. GetCallRetryInfo	12
2.4.5. GetFclass	12
2.4.6. GetDataModulationSupported	13
2.4.7. GetDataProtocol	13
2.4.8. GetDataCompression	14
2.4.9. GetPlusVTRCommandSupported	14
2.4.10. Non-Standard Actions Implemented by a UPnP Vendor	15
2.4.11. Relationships Between Actions	15
2.4.12. Common Error Codes	15
2.5. THEORY OF OPERATION	15
3. XML SERVICE DESCRIPTION	16
4. TEST	20

List of Tables

Table 1: State Variables	5
Table 1.1 : allowedValueList for LinkType	6
Table 1.2 : allowedValueList for DataModulationSupported	6
Table 1.3 : allowedValueList for DataProtocol	7
Table 1.4 : allowedValueList for DataCompression	7
Table 2: Event Moderation	8

Table 3: Actions	9
Table 4: Arguments for SetISPInfo.....	10
Table 5: Arguments for SetCallRetryInfo	10
Table 6: Arguments for GetISPInfo.....	11
Table 7: Arguments for GetCallRetryInfo	12
Table 8: Arguments for GetFclass	12
Table 9: Arguments for GetDataModulationSupported	13
Table 10: Arguments for GetDataProtocol.....	13
Table 11: Arguments for GetDataCompression	14
Table 12: Arguments for GetPlusVTRCommandSupported	14
Table 13: Common Error Codes.....	15

1. Overview and Scope

This service definition is compliant with the UPnP Device Architecture version [1.0](#).

This service-type models physical and link layer properties specific to a Plain Old Telephone System (POTS) modem used for Internet access on an *InternetGatewayDevice**. These properties are common across the different instances of *WANPPPConnection* on the same *WANDevice*.

The service is OPTIONAL (for gateways that support an integrated POTS modem) and is specified in **urn:schemas-upnp-org:device:WANConnectionDevice** one or more instances of which are specified under the device **urn:schemas-upnp-org:device:WANDevice**

An instance of *WANDevice* is specified under the root device **urn:schemas-upnp-org:device:InternetGatewayDevice**

The Service State Table (SST) of this service has dependencies on the *WANPPPConnection* service and must be used in the context of one or more *WANPPPConnection* instances.

NOTE: A *WANDevice* also provides a *WANCommonInterfaceConfig* service that encapsulates Internet access properties common across all WAN interfaces.

1.1. Change Log

Changes from *WANPOTSInterfaceConfig:0.1*

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

Changes from *WANPOTSInterfaceConfig:0.2* as per WC meeting on 9/6/00

- Moved LinkStatus to WANCommonInterfaceConfig
- Define one 'Get' function for all SST variables

Changes from *WANPOTSInterfaceConfig:0.3* as per WC meeting on 10/17/00

- Renames to WANPOTSLinkConfig
- Added ISPPhoneNumber, NumberOfRetries, DelayBetweenRetries

Changes from *WANPOTSLinkConfig:0.4* as per WC meeting on 10/31/00

- Made SetISPInfo optional (because of security concerns)
- Split GetCallSetupInfo action into GetISPInfo(optional) and GetCallRetryInfo (required)

Changes from *WANPOTSLinkConfig:0.5*

- Made GetCallRetryInfo Optional.

Changes from *WANPOTSLinkConfig:0.51*

- Changed Boolean values to 1 and 0.
- Removed white spaces from XML section.
- Changed default value for empty strings to tags with no element values

Changes from *WANPOTSLinkConfig:0.52*

- Modified names of formal parameters of actions to be different from 'Related State Variable'.
- Added state variable LinkType to accommodate non-PPP POTS connections
- Removed 'retval' and empty defaultvalue tags from the XML specification.

Changes from *WANPOTSLinkConfig:0.53*

- Updated to service template v1.01
- Verified against TDC checklist v1.01

* Refer to companion documents defined by the UPnP Internet Gateway working committee for more details on specific devices and services referenced in this document.

- Removed AOL_Dialup from allowed value list for LinkType.
- Fixed Fclass typo in XML section.
- Removed 'Other' as an allowedValue
- Split GetPOTSInterfaceProperties into 5 actions for supporting optional variables.
- Made GET actions of configuration variables required, but set remains optional (due to security constraints)
- Made service optional for vendors who do not want to expose configuration variables

Changes from *WANPOTSLinkConfig:0.8*

- Removed Default values and updated XML section accordingly
- Made Required versus Optional changes to allowedValueList tables
- Changed text description for ISPPhoneNumber
- Deleted Vendor Defined rows in allowedValueList tables

Changes from *WANPOTSLinkConfig:0.81*

- Added XML comment tags to comments text in XML template

Changes from *WANPOTSLinkConfig:0.82*

- Updated tests section
- Deleted allowedvalue range for NumberOfRetries, DelayBetweenRetries and PlusVTRCommandSupported
- Fixed typo with GetDataModulationSupported

Changes from *WANPOTSLinkConfig:0.9*

- Version changes for 1.0
- Changed all occurrences of FClass to Fclass

Changes from *WANPOTSLinkConfig:0.99*

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

Changes from *WANPOTSLinkConfig: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:[WANPOTSLinkConfig: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
ISPPhoneNumber	R	string	N/A	Empty string	N/A
ISPInfo	R	string	N/A	Empty string	N/A
LinkType	R	string	See table 1.1	<i>PPP_Dialup</i>	N/A
NumberOfRetries	R	ui4	>= 0	Not specified	N/A

Variable Name	Req. or Opt. ¹	Data Type	Allowed Value ²	Default Value ²	Eng. Units
DelayBetweenRetries	R	ui4	>= 0	Not specified	seconds
Fclass	O	string	Comma separated sub-strings indicating numbers 0, 1, 2, 2.0, 8, 80	Empty string	N/A
DataModulationSupported	O	string	See table 1.2	Empty string	N/A
DataProtocol	O	string	See table 1.3	Depends on specific modem	N/A
DataCompression	O	string	See table 1.4	Depends on specific modem	N/A
PlusVTRCommandSupported	O	boolean	1, 0	Not specified	N/A
<i>Non-standard state variables implemented by an UPnP vendor go here.</i>	<i>X</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>

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: Most 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 LinkType

Value	Req. or Opt. ³
PPP_Dialup	<u>R</u>

³
—

Table 1.2: allowedValueList for DataModulationSupported

Value	Req. or Opt. ³
V92	<u>O</u>
V90	<u>O</u>
V34	<u>O</u>
V32bis	<u>O</u>

V32	<u>0</u>
-----	----------

3

Table 1.3: allowedValueList for DataProtocol

Value	Req. or Opt. ³
V42_LAPM	<u>0</u>
V42_MNP4	<u>0</u>
V14	<u>0</u>
V80	<u>0</u>

3

Table 1.4: allowedValueList for DataCompression

Value	Req. or Opt. ³
V42bis	<u>0</u>
MNP5	<u>0</u>

3

2.2.1. ISPPhoneNumber

This variable specifies a list of strings separated by semicolon (;), each string representing a phone number to connect to a particular ISP. The digits of the phone number follow the semantics of the ITU E.164 specification. Delimiters such as brackets or hyphens between the digits of a phone number are to be ignored by the gateway. When the gateway receives an action request to initiate a connection, it will try each of these numbers sequentially starting from the first one to the last until a successful connection is made, or all the numbers are exhausted. For each phone number, upon a failed connection attempt the gateway will retry to connect as specified by the state variables `NumberOfRetries` and `DelayBetweenRetries`. This sequence will be interrupted if a connection termination action is received by the gateway (such as `ForceTermination` or `RequestTermination`). In this case the gateway will set the `LastConnectionError` state variable to `ERROR_COMMAND_ABORTED` and return `ConnectionSetupFailed (704)`. If not interrupted, the gateway will return this error only if all the connection attempts fail. In this case the `LastConnectionError` state variable will be set to the appropriate RAS error received during the last failed connection attempt.

2.2.2. ISPInfo

This variable provides information identifying the Internet Service Provider. The format of the string is vendor specific.

2.2.3. LinkType

This variable indicates the type of POTS link used for the dialup connection.

2.2.4. NumberOfRetries

This variable specifies the number of times the gateway should attempt an Internet connection setup before returning error.

2.2.5. DelayBetweenRetries

This variable specifies the number of seconds the gateway should wait between attempts to setup an Internet connection.

2.2.6. Fclass

This variable specifies capabilities of the POTS modem – i.e., if it handles data (0), fax (1,2,2.0), voice (8), DSVD (80).

2.2.7. DataModulationSupported

This variable exposes the modulation standard used for data.

2.2.8. DataProtocol

This variable exposes the protocol standard for data transfers.

2.2.9. DataCompression

This variable exposes the compression technology implemented on the modem.

2.2.10.PlusVTRCommandSupported

This variable is used for full duplex operation with data and voice.

2.2.11.Relationships Between State Variables

The variables in the SST have no dependencies or relationship other than what is mandated by relevant POTS modem standards and protocols.

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
ISPPhoneNumber	No	No	N/A	N/A	N/A
ISPInfo	No	No	N/A	N/A	N/A
LinkType	No	No	N/A	N/A	N/A
NumberOfRetries	No	No	N/A	N/A	N/A
DelayBetweenRetries	No	No	N/A	N/A	N/A
Fclass	No	No	N/A	N/A	N/A
DataModulationSupported	No	No	N/A	N/A	N/A
DataProtocol	No	No	N/A	N/A	N/A
DataCompression	No	No	N/A	N/A	N/A
PlusVTRCommandSupported	No	No	N/A	N/A	N/A
<i>Non-standard state variables implemented by an UPnP vendor go here.</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>

¹ Determined by N, where Rate = (Event)/(N secs).

² (N) * (allowedValueRange Step).

2.3.1. Event Model

None of the variables are evented.

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
SetISPInfo	<u>Q</u>
SetCallRetryInfo	<u>Q</u>
GetISPInfo	<u>R</u>
GetCallRetryInfo	<u>R</u>
GetFclass	<u>Q</u>

GetDataModulationSupported	<u>O</u>
GetDataProtocol	<u>O</u>
GetDataCompression	<u>O</u>
GetPlusVTRCommandSupported	<u>O</u>
<i>Non-standard actions implemented by an UPnP vendor go here.</i>	X

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

2.4.1. SetISPInfo

This action changes the value of variables used to dial up the ISP.

2.4.1.1. Arguments

Table 4: Arguments for SetISPInfo

Argument	Direction	relatedStateVariable
NewISPPhoneNumber	<u>IN</u>	ISPPhoneNumber
NewISPInfo	<u>IN</u>	ISPInfo
NewLinkType	<u>IN</u>	LinkType

2.4.1.2. Dependency on State (if any)

2.4.1.3. Effect on State (if any)

This action modifies variable values, which may be used in subsequent connection setups.

2.4.1.4. 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.

2.4.2. SetCallRetryInfo

This action changes the number of retry attempts if connection setup failures are encountered.

2.4.2.1. Arguments

Table 5: Arguments for SetCallRetryInfo

Argument	Direction	relatedStateVariable
NewNumberOfRetries	<u>IN</u>	NumberOfRetries
NewDelayBetweenRetries	<u>IN</u>	DelayBetweenRetries

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
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. GetISPInfo

This action retrieves values of state variables related to ISP configuration for activation of a connection.

2.4.3.1. Arguments

Table 6: Arguments for GetISPInfo

Argument	Direction	relatedStateVariable
NewISPPhoneNumber	<u>OUT</u>	ISPPhoneNumber
NewISPInfo	<u>OUT</u>	ISPInfo
NewLinkType	<u>OUT</u>	LinkType

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. GetCallRetryInfo

This action retrieves values of state variables concerned with the activation of a connection.

2.4.4.1. Arguments**Table 7: Arguments for GetCallRetryInfo**

Argument	Direction	relatedStateVariable
NewNumberOfRetries	<i>OUT</i>	NumberOfRetries
NewDelayBetweenRetries	<i>OUT</i>	DelayBetweenRetries

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. GetFclass

This action queries modem capabilities.

2.4.5.1. Arguments**Table 8: Arguments for GetFclass**

Argument	Direction	relatedStateVariable
NewFclass	<i>OUT</i>	Fclass

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. GetDataModulationSupported

This action queries the modulation standard used for data.

2.4.6.1. Arguments**Table 9: Arguments for GetDataModulationSupported**

Argument	Direction	relatedStateVariable
NewDataModulationSupported	<i><u>OUT</u></i>	DataModulationSupported

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. GetDataProtocol

This action queries protocol standard used for data transfers.

2.4.7.1. Arguments**Table 10: Arguments for GetDataProtocol**

Argument	Direction	relatedStateVariable
NewDataProtocol	<i><u>OUT</u></i>	DataProtocol

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. GetDataCompression

This action queries the compression technology implemented on the modem.

2.4.8.1. Arguments**Table 11: Arguments for GetDataCompression**

Argument	Direction	relatedStateVariable
NewDataCompression	<i><u>OUT</u></i>	DataCompression

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. GetPlusVTRCommandSupported

This action queries capability for full duplex operation with data and voice.

2.4.9.1. Arguments**Table 12: Arguments for GetPlusVTRCommandSupported**

Argument	Direction	relatedStateVariable
NewPlusVTRCommandSupported	<i><u>OUT</u></i>	PlusVTRCommandSupported

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. 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).

2.4.11. Relationships Between Actions

Other than the pairing of Get and Set actions, these actions have no specific relationships between them.

2.4.12. 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 13: 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	<i>(Specified by UPnP vendor.)</i>

2.5. Theory of Operation

A **WANConnectionDevice** that has a POTS modem interface MAY implement the **WANPOTSLinkConfig** service. The SST variables in this service give information on specific properties of a POTS modem used for WAN Internet access.

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>SetISPInfo</name>
      <argumentList>
        <argument>
          <name>NewISPPhoneNumber</name>
          <direction>in</direction>
          <relatedStateVariable>ISPPhoneNumber</relatedStateVariable>
        </argument>
        <argument>
          <name>NewISPInfo</name>
          <direction>in</direction>
          <relatedStateVariable>ISPInfo</relatedStateVariable>
        </argument>
        <argument>
          <name>NewLinkType</name>
          <direction>in</direction>
          <relatedStateVariable>LinkType</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>SetCallRetryInfo</name>
      <argumentList>
        <argument>
          <name>NewNumberOfRetries</name>
          <direction>in</direction>
          <relatedStateVariable>NumberOfRetries</relatedStateVariable>
        </argument>
        <argument>
          <name>NewDelayBetweenRetries</name>
          <direction>in</direction>
          <relatedStateVariable>DelayBetweenRetries</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>GetISPInfo</name>
      <argumentList>
        <argument>
          <name>NewISPPhoneNumber</name>
          <direction>out</direction>
          <relatedStateVariable>ISPPhoneNumber</relatedStateVariable>
        </argument>
        <argument>
          <name>NewISPInfo</name>
          <direction>out</direction>

```



```

    <relatedStateVariable>ISPInfo</relatedStateVariable>
  </argument>
  <argument>
    <name>NewLinkType</name>
    <direction>out</direction>
    <relatedStateVariable>LinkType</relatedStateVariable>
  </argument>
</argumentList>
</action>
<action>
  <name>GetCallRetryInfo</name>
  <argumentList>
    <argument>
      <name>NewNumberOfRetries</name>
      <direction>out</direction>
      <relatedStateVariable>NumberOfRetries</relatedStateVariable>
    </argument>
    <argument>
      <name>NewDelayBetweenRetries</name>
      <direction>out</direction>
      <relatedStateVariable>DelayBetweenRetries</relatedStateVariable>
    </argument>
  </argumentList>
</action>
<action>
  <name>GetFclass</name>
  <argumentList>
    <argument>
      <name>NewFclass</name>
      <direction>out</direction>
      <relatedStateVariable>Fclass</relatedStateVariable>
    </argument>
  </argumentList>
</action>
<action>
  <name>GetDataModulationSupported</name>
  <argumentList>
    <argument>
      <name>NewDataModulationSupported</name>
      <direction>out</direction>
<relatedStateVariable>DataModulationSupported</relatedStateVariable>
    </argument>
  </argumentList>
</action>
<action>
  <name>GetDataProtocol</name>
  <argumentList>
    <argument>
      <name>NewDataProtocol</name>
      <direction>out</direction>
      <relatedStateVariable>DataProtocol</relatedStateVariable>
    </argument>
  </argumentList>
</action>
<action>
  <name>GetDataCompression</name>

```

```

    <argumentList>
      <argument>
        <name>NewDataCompression</name>
        <direction>out</direction>
        <relatedStateVariable>DataCompression</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
    <name>GetPlusVTRCommandSupported</name>
    <argumentList>
      <argument>
        <name>NewPlusVTRCommandSupported</name>
        <direction>out</direction>
      <relatedStateVariable>PlusVTRCommandSupported</relatedStateVariable>
    </argument>
  </argumentList>
</action>
<!-- Declarations for other actions added by UPnP vendor (if any) go
here -->
</actionList>
<serviceStateTable>
  <stateVariable sendEvents="no">
    <name>ISPPhoneNumber</name>
    <dataType>string</dataType>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>ISPInfo</name>
    <dataType>string</dataType>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>LinkType</name>
    <dataType>string</dataType>
    <allowedValueList>
      <allowedValue>PPP_Dialup</allowedValue>
    </allowedValueList>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>NumberOfRetries</name>
    <dataType>ui4</dataType>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>DelayBetweenRetries</name>
    <dataType>ui4</dataType>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>Fclass</name>
    <dataType>string</dataType>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>DataModulationSupported</name>
    <dataType>string</dataType>
    <allowedValueList>
      <allowedValue>V92</allowedValue>
      <allowedValue>V90</allowedValue>
      <allowedValue>V34</allowedValue>
    </allowedValueList>
  </stateVariable>

```

```

    <allowedValue>V32bis</allowedValue>
    <allowedValue>V32</allowedValue>
  </allowedValueList>
</stateVariable>
<stateVariable sendEvents="no">
  <name>DataProtocol</name>
  <dataType>string</dataType>
  <allowedValueList>
    <allowedValue>V42_LAPM</allowedValue>
    <allowedValue>V42_MNP4</allowedValue>
    <allowedValue>V14</allowedValue>
    <allowedValue>V80</allowedValue>
  </allowedValueList>
</stateVariable>
<stateVariable sendEvents="no">
  <name>DataCompression</name>
  <dataType>string</dataType>
  <allowedValueList>
    <allowedValue>V42bis</allowedValue>
    <allowedValue>MNP5</allowedValue>
  </allowedValueList>
</stateVariable>
<stateVariable sendEvents="no">
  <name>PlusVTRCommandSupported</name>
  <dataType>boolean</dataType>
</stateVariable>
  <!-- Declarations for other state variables added by UPnP vendor (if
any) go here -->
</serviceStateTable>
</scpd>

```

4. Test

No semantic tests have been defined for this service.

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.