Checklist for UPnP™ Standard Service Templates Version 1.01

Specified for UPnP Forum Working Committee use in evaluating Template Design Complete (TDC)

For V1.01 Standard Service Template

Naming conventions and versioning (Service Template Title Page)

___ 1. The Service name is compliant with UPnP naming and versioning conventions
   a) Service name is descriptive of function, < 64 characters.
   b) Capitalizes first letter of each word used in the name.
   c) Draft version number 0.8 - 0.9 is appended to the service name reflecting TDC.
   d) Template file name reflects the ServiceType and version replacing ":" with a space. For example, ServiceType 0.8.

Overview and Scope (Service Template Section 1)

___ 2. The Overview provides a synopsis of the service's function and intended application.
   a) Provides a clear, 3-5 sentence summary of service functionality from an application perspective.
   b) Should list primary functions at a higher level than the action-set (ie; groups of Actions may be described)
   c) Identifies functionality not addressed by this service.
   d) Includes a change log outlining evolution of the design at a high level.

State Variables (Service Template Section 2.2)

____ 3. State Variable names are compliant with UPnP naming conventions
   a) Are descriptive of the state variable's function, < 32 Characters
   b) Capitalize the first letter of each word used in the name
   c) Do not use reserved prefixes (X_ or A_) except as intended

____ 4. State Variable specifications are complete.
   a) Each variable is specified as Required or Optional.
   b) Data types are specified for each variable.
   c) Variables of type A_ARG_TYPE_StateVariable are correctly specified:
      - Same as any other variable except: List n/a (not applicable) for DefaultValue and n/a for Engineering Units.
   d) Engineering units based on SI standard units are specified where they apply. See http://www.ex.ac.uk.cimt/dictunit/dictunit.htm

____ 5. AllowedValues (Recommended) are correctly specified.
   a) AllowedValue's are specified for Number (allowedValueRange) and String (allowedValueList) data types only.
   b) AllowedValues (if specified) are consistent with the bounds set by the variable's data type. For example, a variable of type "i1" must not list an allowedValue exceeding 127.
   c) AllowedValues (if specified) are clearly defined including:
      - Required or Optional values
      - Standard Value or "vendor-defined" placeholders - in the proper word style
      - Footnotes (where necessary) to identify standard requirements.
   d) AllowedValueRange (if specified) includes values or "vendor-defined" placeholders for a range specification including: min, max and step.

___6. DefaultValues (Optional) are correctly specified.

a) DefaultValues (Optional) are specified where necessary to define the initial or reset state of the service.
b) Default values (if specified) are clearly defined including:
    - Required or Optional
    - Standard Value or "vendor-defined" placeholders - in the proper word style
    - Footnotes (where necessary) to identify standard requirements.
c) Default values (if specified) are consistent with the boundaries set by allowedValues and the variable's data type.

____ 7. State Variable descriptions (recommended) adequately define function and usage.

a) Variable Description specifies each variable's run-time persistence (static or dynamic) and functional significance.
b) Relationships between state variables including any functional groupings, or modeling dependencies (e.g. Array modeling) etc. are specified following the state variable descriptions.

Eventing and Moderation (Service Template section 2.3)

___ 8. All state variables are specified in the event moderation table.

a) Exception: Specification of A_ARG_TYPE_StateVariables in this table is optional since variables of this type do not have any Eventing functionality.
b) Variables with evented = "No" list n/a (not applicable) for moderation specifications.
c) Variables with evented = "No" affect a static property in the XML service description, and require changing the XML service schema from the default <stateVariable sendEvents = "Yes"> to "No" for each state variable that is not evented.

___ 9. Moderation specifications are complete

d) Variables with evented = "Yes" may include moderation specifications (optional). Note, in the absence of moderation, events are triggered on each change in the variable's value.
e) All variables that are moderated must specify a moderation effect:
    - Max Rate
    - Min Delta
    - Some combination of the two: Max Rate And/OR Min Delta

___ 10. The Description of the Event Model (recommended) should summarize basic design assumptions and requirements for eventing and moderation for use by an application developer including:

a) User interface requirement (if any)
b) Asynchronous action requirement (if any)
c) Functional versus Max Rate tradeoffs for moderation (if any)
d) If a relatively high event frequency is required, an estimate for the maximum event rate associated with the service?
e) Reason not evented (if not obvious)

Actions (Service Template section 2.4)

___ 11. Action names are compliant with UPnP naming conventions
- a) All actions are listed in Table 3 as either Required or Optional.
- b) Action names are descriptive of the function, and are < 32 characters.
- c) Action names are in "Verb-Noun" form (recommended) For example, ActionObject = SetTargetLevel).
- d) The first letter of each word used in each action name is capitalized.
- e) Actions do not use reserved prefixes (e.g. X_, A_ etc.) except as intended. (X_ is for non standard actions, state variables, allowed Values etc.; A_ is reserved by the UPnP architecture – for example, A_ARG_TYPE_StateVariables).

___ 12. Action specifications are complete as necessary to facilitate implementation and test.
- a) A complete Action Specification is provided for each action supported by the service in section 2.4.x.
- b) For each action, a description (recommended) of the action follows the section header, "2.4.x ActionName" as needed to supplement detailed action specifications.
- c) Action descriptions describe the functional behavior of the action, for example, what the action does; the need for "Atomic Read or Write access to variable sets" etc.
- d) A Description of the Relationship Between Actions (following the last Action Specification in section 2.4) is provided as necessary to describe dependencies or interactions between actions in a service including shared state variables, atomic action sequences etc.

___ 13. Actions are defined for synchronous operation
- a) Actions are guaranteed to complete by returning response codes and any "Out" arguments in < 30 seconds.

___ 14. Action argument specifications are complete
- a) Argument names follow the same naming conventions as state variables – see 3. above. Note; all action arguments specified must be implemented by the service.
- b) Action argument names are unique from StateVariable names. (ie; Action arguments are different from state variables). For example, arguments cannot be queried via a Get( ) action, do not persist state etc.
- c) A direction (In, Out) with respect to the service has been specified for each argument. For example, In-arguments are passed to a service when an action is invoked, while out arguments return values as a result of the action.
- d) "IN" arguments are listed first before "OUT" arguments in the Action Specification.
- e) The first and only the first "OUT" argument for each action may include a footnote[R] to declare a "retval" for the action. This declaration is optional and is provided for platform APIs that expose a return value related to this action.

___15. A RelatedStateVariable is specified for each argument.

a) A RelatedStateVariable of the same data type is specified for each of the action's arguments as necessary to type the argument.
b) If a RelatedStateVariable of the appropriate DataType did not exist, one has been declared with prefix A_ARG_TYPE where Type may be any of UPnP supported data types.
c) All relatedStateVariables of type A_ARG_TYPE must be specified in "Table 1 State Variables" in accordance with 4. c) above.

___16. The action's behavior in the context of service state has been specified as necessary to support definition of semantic test cases.

a) An action's dependency on state (if any) has been specified including:
    - State preconditions (if any) affecting the actions behavior where such behavior must be consistent for all UPnP services of this type.
    - Description of how state preconditions affect the action's behavior.
    - Alternately, this section may reference section 2.5, Theory of Operation to provide a service wide view of action-state dependencies.
b) The Action's Effect on Service State (if any) has been specified including:
    - Changes in service state resulting from action invocation.

___ 17. All Error Codes that apply have been specified for each action

a) All common error codes that apply are listed for each action including 401-403, 501, and 600-699 (common action errors) – See the service template, Common Error codes for a complete listing for this service..
b) All action-specific errors (700-799) that apply to standard service actions are listed in the errorCode specifications for each action with a description of the error. (Note; action-specific 7xx errors may also be listed for reference in the Common Error Codes table of the Service Standard.)
c) Action errors that are dependent on service state (if any) include a description of state dependencies.
d) Each Error Description is provided in the context of the end user and should be less than 256 characters.

Theory of operation (Service Template section 2.5)

___ 18. The Theory of operation (optional) provides a description of service functionality as necessary to facilitate implementation and application of this service.

a) Includes definition of terms where necessary.
b) Provides a pseudo code description of action sequences that demonstrate how a control point is intended to interact with this service.
c) Provides a description of the internal function of the service for example, it's state model (if applicable).

XML Service Template (Service Template section 3.0)

___ 19. The XML Service Template is complete
   a) Information identified by Red italics has been specified by the working committee.
   b) The XML template is consistent with all table specifications for State Variables, Evented Variables, Actions and Arguments defined in section 2 of the draft standard.
   c) All optional state variables and actions have been specified in the XML service template.
   d) Placeholders for "vendor-defined" allowedValues are specified in the XML template.

___ 20. The XML syntax is well formed. Use the following procedure to verify syntax:
   a) Procedure to be provided.

___ 21. The XML schema is valid in accordance with the UPnP template language. Use the following procedure to validate the schema:
   a) Procedure to be provided.

Test (Service Template section 4.0)

___ 22. TBD

Template Design Complete (TDC) - to be formally declared by the Working Committee

___ 23. This Service template meets Version .8 TDC criteria suitable for implementation and test.
   a) This service standard supports targeted device application scenarios.
   b) There are no unresolved design issues that would prevent sample implementations.
   c) The design has been reviewed by at least 3 sample implementers.
   d) The service model is well defined in accordance with this checklist.
   e) This service standard effectively balances tradeoffs between:
      - Baseline functional requirements to be exposed in V1 services
      - Implementation complexity (no. of state variables and actions)
      - Re-usability (modular, generic building blocks where feasible)
      - Extensibility (Is extensible for Version 2 of the service – if applicable).
   f) Service options (actions and state variables) are limited to the core set committed by sample implementers for standardization.
   g) The XML service description is complete.
   h) Optional and recommended modeling specifications have been completed to the satisfaction of the working committee including:
      - AllowedValues
      - DefaultValues
      - Variable descriptions including relationship between state variables
      - Event moderation
      - Action descriptions, dependency on state, and effect on state
      - Action error codes
      - Theory of operation