

PG483 - Ubiquitous Media
Anhang
textit

Oleg Andreev
Daniel Arndt
Jan Beisenkamp
Marcus Rogério Blitz
Martin Had
Da He
Christoph Fiehe
Julian Flake
David Kliczbor
Nikinaz Nazemi
Mario Wündsche
Maxim Zaks

28. September 2006

1 Anhang

1.1 Klassen-Dokumentation

Eine umfangreiche Dokumentation der Klassen des entwickelten ControlPoints kann anhand der nachfolgenden Abschnitte entnommen werden. Die Grundlage der aufgeführten Daten basiert auf den entwickelten Gegebenheiten und kann als Basissatz für weitere Entwicklungen angesehen werden. Die Gliederung erfolgt anhand der jeweiligen Klassen und wird durch eine Auflistung der Methoden, Events und Variablen ergänzt.

1.1.1 Namespace: UniDo.PG483.CompactCP

Klasse: AVSession Die Klasse „AVSession“ beschreibt die logische Abstraktion einer Verbindung, bestehend aus Quelle (Server), Senke (Renderer) und Medienobjekt. Jede solche virtuelle Verbindung bzw. Session wird hier als ein Objekt der Klasse „AVSession“ betrachtet, welches eindeutig mit diesen drei Komponenten verbunden ist.

Dieses führt weiterhin alle von UPnP vorgesehenen Steuerungs- und Koordinationsprozesse aus, wodurch die Kontrolle individueller Medien-Wiedergaben ermöglicht wird, unabhängig davon, ob zunächst eine neue Verbindung initiiert, oder eine bereits vorhandene Verbindung gesteuert werden soll. Ebenso übernimmt es die Kommunikation mit den beteiligten Parteien, genauer gesagt, mit den Aktionen der Dienste, auf Server- und Renderer-Seite.

Da maximale Unterstützung gewährleistet werden soll, ist diese Implementierung auch in der Lage, mit mehreren Verbindungen auf einem Gerät umzugehen. Die sich daraus ergebenden Schwierigkeiten bei der Berücksichtigung aller theoretisch denkbaren Szenarien erschweren ein schnelles Verständnis dieser Klasse und ihrer Funktionsweise.

An dieser Stelle sei erwähnt, dass die folgenden Abläufe nur mit einem umfassenden Verständnis des UPnP-Protokolls nachvollzogen werden können. Es ist daher dringend anzuraten, die UPnP-Dokumentation zu den Diensten „AVTransport“ ?, „ConnectionManager“ ? und „RenderingControl“ ? im Vorfeld gelesen zu haben.

Event: OnConnectionEstablishedToExistingAVSession Eingabe: AVSession avSession, UPnPDevice server, UPnPDevice renderer
Ausgabe: void

Dieses Event wird geworfen, um den AVSession-Manager über den erfolgreichen Verbindungsaufbau zu einer bereits existierenden aber nicht selbst initiierten AVSession zu unterrichten. Der Parameter „avSession“ verweist auf die neue AVSession, die Variablen „server“ und „renderer“ bezeichnen jeweils – falls bekannt – den an der Verbindung beteiligten Server und Renderer. Durch den Aufruf dieses Events, wird die Methode

„ConnectingToExistingAVSession“ im AVSessionManager-Objekt aufgerufen.

Event: OnErrorOccurred Eingabe: AVSession avSession, UPnPInvokeException ex
Ausgabe: void

Sollte es zu einem Fehler innerhalb eines der vielen Methodenaufrufen kommen, so kann dieses universelle Event verwendet werden, um den AVSessionManager über das spezielle Problem zu informieren. Durch den Aufruf dieses Events, wird die Methode „ErrorOccurredOnAVSessionSink“ im AVSessionManager-Objekt aufgerufen. Der Parameter „avSession“ verweist auf die AVSession-Instanz, in der der Fehler aufgetreten ist.

Event: OnNewConnectionEstablished Eingabe: AVSession avSession, UPnPDevice server, UPnPDevice renderer, Uri currentURI, CDSObject cdsObject, bool toPlay
Ausgabe: void

Wurde vom Nutzer aus eine neue Verbindung initiiert, so wird durch den Aufruf dieses Events das „AVSessionManager“-Objekt darüber informiert. Übergeben wird eine Referenz auf die neu erstellte „AVSession“-Instanz, sowie die an dieser AVSession beteiligten Hauptkomponenten in Form der Variablen „server“, „renderer“ und „cdsObject“. Der Parameter „currentURI“ bezeichnet die verwendete „AVTransportURI“ und die boolesche Variable „toPlay“ gibt an, ob zusätzlich noch ein „Play“-Befehl zur Initiierung des Abspielvorgangs an den Server oder den Renderer zu schicken ist. Durch den Aufruf dieses Events, wird die Methode „NewConnectionEstablishedSink“ im AVSessionManager-Objekt aufgerufen.

Methode: AVSession Sichtbarkeit: public

Eingabe: UPnPDevice device, Int32 connectionID, ConnectionEstablishedToExistingAVSession callbackOnConnectionEstablished, ErrorOccurred callbackOnErrorOccurred

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor, mit dessen Hilfe ein „AVSession“-Objekt erzeugt wird. Im Sinne von UPnP kann nun eine Medienwiedergabe eingeleitet werden. Dieser Konstruktor wird dann verwendet, wenn eine bestehende Medienwiedergabe in Form einer in UPnP-spezifizierten ConnectionID durch ein „AVSession“-Objekt gekapselt werden muss. Der Parameter „device“ beschreibt das Device, zu dem unter Verwendung der angegebenen „ConnectionID“ eine Verbindung hergestellt werden soll. Die beiden Methoden-Zeiger „callbackOnConnectionEstablished“ und „callbackOnErrorOccurred“ werden verwendet, um im Falle einer erfolgten bzw. nicht erfolgten Verbindung die beiden Methoden „ConnectingToExistingAVSession“ bzw. „ErrorOccurredOnAVSessionSink“ im AVSession-Manager aufzurufen. Durch den Aufruf der Methode „GetCurrentConnectionInfoRenderer“ werden Informationen über die Verbindung eingeholt, die durch den Parameter „ConnectionID“ beschrieben werden.

Methode: AVSession Sichtbarkeit: public

Eingabe: UPnPDevice device, String useProtocol, Uri avTransportURI, NewConnectio-

nEstablished callbackOnConnectionEstablished, ErrorOccurred callbackOnErrorOccurred

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor, mit dessen Hilfe ein „AVSession“-Objekt erzeugt wird. Verwendung findet dieser Konstruktor nur, falls eine neue AVSession durch die im Follow-Me-Modus befindliche AVSession aufgebaut werden soll, ausgelöst durch einen veränderten Aufenthaltsort des Nutzers. Das „device“-Objekt bezeichnet den Renderer, zu dem die neue AVSession etabliert werden soll, „useProtocol“ bezeichnet das zu verwendende Protokoll, „avTransportURI“ die zu verwendende Medienposition im Netzwerk. Die beiden Methoden-Zeiger „callbackOnConnectionEstablished“ und „callbackOnErrorOccurred“ werden verwendet, um im Falle einer erfolgten bzw. nicht erfolgten Verbindung die beiden Methoden „NewConnectionEstablishedSink“ bzw. „ErrorOccurredOnAVSessionSink“ im AVSession-Manager aufzurufen. Da bereits das zu verwendende Protokoll sowie die „AVTransportURI“ bekannt sind, erfolgt im Anschluss der Aufruf der Methode „PrepareForConnectionRenderer“.

Methode: AVSession Sichtbarkeit: public

Eingabe: UPnPDevice server, UPnPDevice renderer, CDSObject cdsObject, NewConnectionEstablished callbackOnConnectionEstablished, ErrorOccurred callbackOnErrorOccurred

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor, mit dessen Hilfe ein „AVSession“-Objekt erzeugt wird. Wählt der Nutzer nach einem „Browse“ oder „Search“ ein abzuspielendes Medien-Objekt aus, so wird dieser Konstruktor verwendet. In diesem Szenario stehen alle bekannten und wünschenswerten Informationen über alle an einer AVSession beteiligten Komponenten zur Verfügung, die innerhalb des „AVSession“-Objektes gespeichert werden. In den beiden Structs „ConnectionManagerServerInfo“ und „ConnectionManagerRendererInfo“ können nun die Information über die verwendeten „ConnectionManager“-Objekte gespeichert werden. Im Anschluss erfolgt ein Aufruf der Methode „GetProtocolInfo“, die Informationen über die von einem Renderer unterstützen Protokolle einholt.

Methode: GetCompatibleProtocols Sichtbarkeit: private

Eingabe: UPnPDevice device, String[] protocols

Ausgabe: void

Nach einem erfolgreichen Aufruf der Methode „GetProtocolInfo“, wird aus der Callback-Methode namens „GetProtocolInfoSink“ diese Methode gerufen. Übergeben werden das Renderer-Objekt, sowie die von diesem Gerät unterstützten Transfer-Protokolle. Die Aufgabe dieser Methode ist es, zwischen den Transfer-Protokollen des Medien-Objekts und denjenigen auf Seiten des Renderers einen Abgleich vorzunehmen, um so ein passendes auszuwählen. Die vom Medien-Server unterstützen Protokolle sind bereits bekannt, da sie nach der erfolgten „Browse“- bzw. „Search“-Aktion bereits vom Server mitgeliefert werden und in der Variablen „Res“ in der CDSObject-Instanz gespeichert worden

sind. Ein Abgleich der Protokolle wird unter Anwendung eines regulären Ausdrucks vorgenommen. Im Falle eines erfolgten Matches, werden in die Listen „possibleProtocols“ auf Server- und auf Renderer-Seite jeweils die verwendbaren Protokolle eingetragen. In den Variablen „possibleProtocols“ sind also alle potentiell möglichen Protokoll-Paare gelistet. Durch eine Index-Auswahl innerhalb einer der beiden Listen steht unter dem gleichen Index der anderen Liste das passender Gegenstück zum ausgewählten Protokoll zur Verfügung. Es sind daher alle praktisch möglichen Permutationen aller Protokolle innerhalb dieser Liste vorhanden. Die Variablen „useProtocol“ auf beiden Seiten erhält nun das tatsächlich ausgewählte Protokoll. An dieser Stelle wurde vereinfacht das Protokoll an Index-Position 0 verwendet, falls mindestens ein kompatibles Protokoll zur Verfügung steht. Stattdessen wäre eine GUI-Ausgabe denkbar, die den Nutzer auffordert ein von ihm präferiertes Protokoll auszuwählen. Ebenso wäre eine prioritätsbasierte automatische Auswahl möglich. Hier besteht also noch Optimierungsspielraum.

Methode: GetCurrentConnectionInfoRenderer Sichtbarkeit: private

Eingabe: UPnPDevice device, Int32 ConnectionID

Ausgabe: void

Diese Methode wird verwendet, wenn beim Eintritt des Kontrollpunktes in das UPnP-Netzwerk eine bereits laufende Medien-Wiedergabe erkannt wurde. Das Wiedergabegerät wird durch den Aufruf der Methode „GetCurrentConnectionInfo“ kontaktiert, um so die zu der aktuell laufenden Verbindung passenden IDs zur Steuerung zu erhalten.

Methode: GetCurrentConnectionInfoRendererSink Sichtbarkeit: private

Eingabe: ConnectionManager sender, Int32 ConnectionID, Int32 RcsID, Int32 AVTransportID, String ProtocolInfo, String PeerConnectionManager, Int32 PeerConnectionID, ConnectionManager.Enum_A_ARG_TYPE_Direction Direction, ConnectionManager.Enum_A_ARG_TYPE_ConnectionStatus Status, UPnPInvokeException ex, object _Tag

Ausgabe: void

Diese Methode wird als Callback bei einem erfolgreich durchgeführten Aktionsaufruf „GetCurrentConnectionInfo“ aufgerufen. Sie liefert alle zu einer Verbindung benötigten Steuerungs-Informationen. Da die Methode „GetCurrentConnectionInfo“ in der Methode „GetCurrentConnectionInfoRenderer“ gerufen wird, stellen die Eingabe-Parameter, die Rückgabe-Werte bezüglich einer speziellen Verbindung seitens des – an der Verbindung beteiligten – Renderers dar. Aufgrund einer gewissen Liberalität bzgl. Eingabe-Parameter kann nicht vorausgesetzt werden, dass die „PeerConnectionID“, sowie der „PeerConnectionManager“ gesetzt sind. Daher kann an dieser Stelle nicht trivialerweise auf die Quelle, genauer gesagt den Server, innerhalb einer Verbindung geschlossen werden. Hat man es mit einer im Sinne von UPnP definierten Streaming-Sitzung zu tun, so wird notwendigerweise die AVTransport-Instanz auf Server-Seite benötigt. Der kontaktierte Renderer wird als Rückgabewert eine AVTransportID von –1 liefern, die signalisiert, dass der AVTransport-Dienst auf Renderer-Seite innerhalb der angegebenen Verbindung nicht angeboten wird. Wird nun entdeckt, dass keine gültige AVTransportID

vorhanden ist, so wird – falls die „PeerConnectionID“ und der „PeerConnectionManager“ gesetzt sind – eine Verbindung zu dem dadurch angegebenen Server hergestellt, um eine gültige Instanz von der Server-Seite zu erhalten. Schlägt dies fehl, so muss versucht werden, eine gültige AVTransport-Instanz von der Server-Seite zu erhalten. Da keine Information über den Quell-Server zur Verfügung steht, bleibt keine andere Wahl, als jeden bekannten Server zu kontaktieren und für jede bekannte „ConnectionID“ nähere Informationen einzuholen. Da beim Entwurfszeitpunkt die Schwierigkeiten bzgl. nicht-selbst initiiertes Verbindungen nicht bedacht wurden, ist die vorhandene Funktionalität nachträglich eingebaut worden und fügt sich nur sehr schlecht in die Implementierung ein. Es ist daher anzuraten, diese Stelle zu überarbeiten und neue Informationsflüsse zu definieren, die eine „saubere“ Verwaltung bestehender serverseitiger Verbindungen erlauben. Da sichergestellt werden muss, dass die LastChange-Nachrichten von Seiten des Renderers, die des Server nicht überholen, werden an dieser Stelle die ConnectionIDs der Server synchron angefordert. Auch diese Vorgehensweise sollte in einer späteren Version nach Möglichkeit vollständig ersetzt werden.

Diese Stelle zeigt deutlich eine Schwäche der erstellten Implementierung auf. Sinnvoll wäre es, die Session-Verwaltung als ein Matching-Problem aufzufassen. Alle Informationen aller Server und Renderer werden entgegengenommen und gespeichert. Ein Memorymap-Prinzip könnte somit den vorhandenen Ansatz ersetzen. Dabei geht es darum, aus den vorhandenen Informationen ein Paar bestehend aus Server und Renderer wieder zu vereinen. Ergibt sich ein solches Match zwischen den empfangenen Informationen vom Renderer und Server, so sollte eine AVSession erzeugt werden, die sich vollständig steuern lässt. Erst bei einer sauberen Paarbildung wird die Steuerung dieser Session freigegeben. Allerdings ist dieses Prinzip enorm streng und stellt Bedingungen an das UPnP-Protokoll, die nicht von allen standardkonformen Implementierungen eingehalten werden. Zum Implementierungszeitpunkt stand kein Server zur Verfügung, der ein solches oben angesprochenes Szenario unterstützt. Daher konnte diese Teil-Implementierungen nicht auf korrekte Funktionalität getestet werden.

Methode: GetCurrentConnectionInfoServer Sichtbarkeit: private

Eingabe: UPnPDevice device, Int32 connectionID, bool idKnown

Ausgabe: void

Die Methode fordert zu einer angegebenen „connectionID“ auf einem Server-Device nähere Informationen an. Das Flag „idKnown“ wird auf true gesetzt, falls die „PeerConnectionID“ von Seiten des Renderers ausgelesen werden konnte und daher der Quell-Server eindeutig identifiziert ist.

Methode: GetCurrentConnectionInfoServerSink Sichtbarkeit: private

Eingabe: ConnectionManager sender, Int32 ConnectionID, Int32 RcsID, Int32 AVTransportID, String ProtocolInfo, String PeerConnectionManager, Int32 PeerConnectionID, ConnectionManager.Enum_A_ARG_TYPE_Direction Direction,

ConnectionManager.Enum_A_ARG_TYPE_ConnectionStatus Status,

UPnPInvokeException ex, object _Tag

Ausgabe: void

Diese Methode wird als Callback bei einem erfolgreich durchgeführten Aktionsaufruf „GetCurrentConnectionInfo“ aufgerufen. Sie liefert alle zu einer Verbindung benötigten Steuerungs-Informationen. Da die Methode „GetCurrentConnectionInfo“ in der Methode „GetCurrentConnectionInfoServer“ gerufen wird, stellen die Eingabe-Parameter die Rückgabe-Werte bzgl. einer speziellen Verbindung seitens des an der Verbindung beteiligten Servers dar. Die in der Variablen „_Tag“ getunnelte Information enthält ein Flag, das angibt, ob die „PeerConnectionID“ des Servers zuvor bekannt war oder nicht. Ist sie bekannt, so muss der Server eine AVTransport-Instanz anbieten, mit deren Hilfe eine Steuerung der AVSession möglich ist. Im negativen Fall kann keine Verbindung hergestellt werden. Sollte die „PeerConnectionID“ nicht bekannt sein, so wird für jeden Server ein Abgleich vorgenommen, ob eine dort bestehende Verbindung mit dem zuvor kontaktierten Renderer übereinstimmt, indem die übermittelte PeerConnectionID vom Server verglichen wird mit der „ConnectionID“ des Renderers, die zuvor in der Variablen „connectionID“ im Struct „connectionManagerRendererInfo“ gespeichert wurde. Sollte eine passendere serverseitige Verbindung gefunden werden, erfolgt eine Eintragung in das Struct „connectionManagerServerInfo“, sowie in „avTransportInfo“. Nun kann die Steuerung einer zuvor etablierten Sitzung übernommen werden, was durch das Event „OnConnectionEstablishedToExistingAVSession“ signalisiert wird.

Methode: GetProtocolInfo Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Mit Hilfe dieser Methode werden die vom Renderer unterstützen Transfer-Protokolle abgerufen, indem die Methode „GetProtocolInfo“ des zum Renderer gehörigen ConnectionManager-Objektes gerufen wird. Als Callback wird die Methode „GetProtocolInfoSink“ gerufen, die die Rückgabe-Werte entgegen nimmt.

Methode: GetProtocolInfoSink Sichtbarkeit: private

Eingabe: ConnectionManager sender, String Source, String Sink, UPnPInvokeException ex, object _Tag

Ausgabe: void

Diese Methode wird als Callback bei einem erfolgreich durchgeführten Aktionsaufruf „GetProtocolInfo“ aufgerufen. Sie liefert alle vom Renderer unterstützen Transfer-Protokolle, die das Gerät als Quelle bzw. Senke unterstützt. Die beiden Parameter „Source“ und „Sink“ enthalten nun die Komma-separierten Protokolle. Durch den Aufruf von „GetCompatibleProtocols“ werden alle zu diesem Renderer kompatiblen Protokolle ermittelt, die von der Server-Seite für einen Medientransfer angeboten werden. Diese Methode kann nur in dem Szenario gerufen werden, wenn eine vom User initiierte Session gestartet werden soll, da ansonsten nicht alle vom Server angebotenen Quell-Protokolle für den Transfer des zuvor gewählten Mediums verwendet werden können.

Methode: Pause Sichtbarkeit: public

Eingabe: Pausing callbackPausing

Ausgabe: void

Diese Methode versetzt den Abspielzustand des Mediums, das mit dieser AVSession verbunden ist, in den Pause-Zustand. Der angegebene Methodenzeiger, wird verwendet, um einen erfolgreichen Wechsel in diesen neuen Zustand zu signalisieren.

Methode: PauseSink Sichtbarkeit: private

Eingabe: AVTransport sender, UInt32 InstanceID, UPnPInvokeException ex, object _Tag

Ausgabe: void

Diese Methode wird aufgerufen, sobald die UPnP-AV-Implementierung erfolgreich oder nicht erfolgreich die „Pause“-Nachricht abgesetzt hat. Die Variable „sender“ beschreibt dabei die zur Steuerung aufgerufene „AVTransport“-Instanz, „InstanceID“ bezeichnet die exakte ID der Verbindung, die zur Steuerung auf Server- oder Renderer-Seite verwendet wurde.

Methode: Play Sichtbarkeit: public

Eingabe: Playing callbackPlaying

Ausgabe: void

Diese Methode versetzt den Abspielzustand des Mediums, das mit dieser AVSession verbunden ist, in den Play-Zustand. Der angegebene Methodenzeiger, wird verwendet, um einen erfolgreichen Wechsel in diesen neuen Zustand zu signalisieren.

Methode: PlaySink Sichtbarkeit: private

Eingabe: AVTransport sender, UInt32 InstanceID, AVTransport.Enum.TransportPlaySpeed Speed, UPnPInvokeException ex, object _Tag

Ausgabe: void

Diese Methode wird aufgerufen, sobald die UPnP-AV-Implementierung erfolgreich oder nicht erfolgreich die „Play“-Nachricht abgesetzt hat. Die Variable „sender“ beschreibt dabei die zur Steuerung aufgerufene „AVTransport“-Instanz, „InstanceID“ bezeichnet die exakte ID der Verbindung, die zur Steuerung auf Server- oder Rendererseite verwendet wurde.

Methode: PrepareForConnectionRenderer Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Diese Methode führt den in UPnP-spezifizierten Ablauf zum Verbindungsaufbau auf Seiten des Renderers durch. Es muss überprüft werden, ob die „PrepareForConnection“-Aktion dort verfügbar ist. Sollte dies der Fall sein, so muss sie verwendet werden, um eine passende ConnectionID, AVTransportID und RenderingControlServiceID zu erhalten. Streng nach Standard wird die Quelle der Verbindung – der Server – durch die „PeerConnectionID“ und den „PeerConnectionManager“ angegeben, falls diese bekannt

sind. Sollte nun weder der Server noch der Renderer, die Aktion „PrepareForConnection“ anbieten, so sind die zu verwendenden IDs nach UPnP jeweils gleich 0. Es kann nun die eigentliche AVTransportURI übermittelt werden, indem die Methode „SetAVTransportURI“ gerufen wird. Sollte nur der Renderer die „PrepareForConnection“-Aktion nicht anbieten, so können anhand der vom Server eingeholten Informationen die zu verwendenden IDs gesetzt und gespeichert werden, sowie die Zugehörigkeiten zur Steuerung eindeutig bestimmt werden. Durch den parametrisierten Aufruf der Aktion „SetAVTransportURI“, der angibt, ob zusätzlich noch ein „Play“-Befehl zu schicken ist, kann nun fortgefahren werden.

Methode: PrepareForConnectionRendererSink Sichtbarkeit: private

Eingabe: ConnectionManager sender, String RemoteProtocolInfo, String PeerConnectionManager, Int32 PeerConnectionID, ConnectionManager.Enum_A_ARG_TYPE_Direction Direction, Int32 ConnectionID, Int32 AVTransportID, Int32 RcsID, UPnPInvokeException ex, object _Tag

Ausgabe: void

Bietet nur der Renderer die „PrepareForConnection“-Aktion an, so wird das Ergebnis dieses Aufrufs in diese Methode geleitet. Anhand der vorliegenden Informationen können nun die IDs gesetzt werden und die Zugehörigkeiten bzgl. Steuerungsereignissen gespeichert werden. Abgeschlossen durch den Aufruf der Methode „SetAVTransportURI“ wird der erste große Teil der gegenseitigen Bekanntgabe abgeschlossen. Zum jetzigen Zeitpunkt kann die vorliegende AVSession eindeutig gesteuert werden.

Methode: PrepareForConnectionServer Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Wurde ein kompatibles Transferprotokoll gefunden (siehe Methode: GetCompatibleProtocols), so wird – falls vorhanden – die „PrepareForConnection“-Aktion auf Serverseite aufgerufen. Steht diese Aktion nicht zur Verfügung, so wird durch den Aufruf der Methode „PrepareForConnectionRenderer“ auf Rendererseite fortgefahren.

Methode: PrepareForConnectionServerSink Sichtbarkeit: private

Eingabe: ConnectionManager sender, String RemoteProtocolInfo, String PeerConnectionManager, Int32 PeerConnectionID, ConnectionManager.Enum_A_ARG_TYPE_Direction Direction, Int32 ConnectionID, Int32 AVTransportID, Int32 RcsID, UPnPInvokeException ex, object _Tag

Ausgabe: void

Sollte der Server die „PrepareForConnection“-Aktion anbieten, so wird diese Methode als Callback gerufen. Es wird eine Vorbelegung der verschiedenen IDs sowie ihrer Zugehörigkeiten durchgeführt. Durch den Aufruf der Methode „PrepareForConnectionRenderer“ wird auf der Seite des Renderers fortgefahren.

Methode: Seek Sichtbarkeit: public

Eingabe: String seekToPosition, Seeking callbackSeeking

Ausgabe: void

Diese Methode setzt die relative Abspielzeit einer Medienwiedergabe auf die durch den Parameter „seekToPosition“ angegebene Zeit. Dieser Parameter hat das Format „hh:mm:ss“. Der angegebene Methodenzeiger, wird als Callback verwendet, welcher aufgerufen wird, sobald eine Antwort, die die abgesetzte Anfrage betrifft, eingetroffen ist.

Methode: SeekSink Sichtbarkeit: private

Eingabe: AVTransport sender, UInt32 InstanceID,

AVTransport.Enum_A_ARG_TYPE_SeekMode Unit, String Target, UPnPInvokeException ex, object _Tag

Ausgabe: void

Diese Methode wird aufgerufen, sobald die UPnP-AV-Implementierung erfolgreich oder nicht erfolgreich die „Seek“-Nachricht abgesetzt hat. Die Variable „sender“ beschreibt dabei die zur Steuerung aufgerufene „AVTransport“-Instanz, „InstanceID“ bezeichnet die exakte ID der Verbindung, die zur Steuerung auf Server- oder Rendererseite verwendet wurde, „SeekMode“ beschreibt den laut UPnP-AV-Spezifikation verwendeten Spulmodus und „Target“ beschreibt die neu gesetzte relative Abspielzeit.

Methode: SetAVTransportURI Sichtbarkeit: private

Eingabe: bool toPlay

Ausgabe: void

Durch den Aufruf dieser Methode wird an den Server oder den Renderer, die zu verwendende AVTransportURI übermittelt. Um sicherzustellen, dass die AVTransportURI auch tatsächlich gesetzt ist, erfolgt ein Abgleich mit dem zuvor ermittelten Transportprotokoll. Auf diese Weise können Inkonsistenzen vermieden werden.

Methode: SetAVTransportURI Sichtbarkeit: public

Eingabe: Uri uri

Ausgabe: void

Methode: SetAVTransportURISink Sichtbarkeit: private

Eingabe: AVTransport sender, UInt32 InstanceID, String CurrentURI, String CurrentURIMetaData, UPnPInvokeException ex, object _Tag

Ausgabe: void

Nachdem die Nachricht mit der zu verwendenden AVTransportURI erfolgreich oder nicht erfolgreich abgeschickt wurde, wird diese Methode von der UPnP-AV-Implementierung gerufen. Diese Methode stellt den vorletzten Schritt bei der Etablierung einer Medienwiedergabe dar. Durch das Werfen des Events „OnNewConnectionEstablished“ signalisiert das „AVSession“-Objekt dem AVSessionManager, dass eine Verbindung erfolgreich hergestellt werden konnte.

Methode: SetVolumeTo Sichtbarkeit: public

Eingabe: Int32 volume, VolumeChanged callbackVolumeChanged

Ausgabe: void

Diese Methode modifiziert die Lautstärke des durch das „AVSession“-Objekt gekapselten Mediums auf das durch den Parameter „volume“ angegeben Level. Der Methodenzeiger „callbackVolumeChanged“ wird verwendet, um zu signalisieren, dass die Lautstärke auf das gewünschte Level gesetzt werden konnte.

Methode: SetVolumeToSink Sichtbarkeit: private

Eingabe: RenderingControl sender, UInt32 InstanceID, RenderingControl.Enum_A_ARG_TYPE_Channel Channel, UInt16 DesiredVolume, UPnPInvokeException ex, object _Tag

Ausgabe: void

Nachdem die Nachricht mit dem neuen Lautstärke-Level erfolgreich oder nicht erfolgreich abgeschickt wurde, wird diese Methode von der UPnP-AV-Implementierung gerufen. Die Variable „sender“ beschreibt dabei die zur Steuerung aufgerufene „RenderingControlService“-Instanz, „InstanceID“ bezeichnet die exakte ID der Verbindung, die zur Steuerung auf Server- oder Rendererseite verwendet wurde, „Channel“ stellt den Kanal dar, wobei in dieser Implementierung nur „Master“ unterstützt wird. Der Parameter „DesiredVolume“ verweist auf das zur Zeit vorherrschende Lautstärke-Level.

Methode: Stop Sichtbarkeit: public

Eingabe: Stopped callbackStopped

Ausgabe: void

Diese Methode versetzt den Abspielzustand des Mediums, das mit dieser AVSession verbunden ist, in den Stop-Zustand. Der angegebene Methodenzeiger wird verwendet, um einen erfolgreichen Wechsel in diesen neuen Zustand zu signalisieren.

Methode: StopSink Sichtbarkeit: private

Eingabe: AVTransport sender, UInt32 InstanceID, UPnPInvokeException ex, object _Tag

Ausgabe: void

Diese Methode wird aufgerufen, sobald die UPnP-AV-Implementierung erfolgreich oder nicht erfolgreich die „Stop“-Nachricht abgesetzt hat. Die Variable „sender“ beschreibt dabei die zur Steuerung aufgerufene „AVTransport“-Instanz, „InstanceID“ bezeichnet die exakte ID der Verbindung, die zur Steuerung auf Server- oder Rendererseite verwendet wurde.

Klasse: AVSessionManager Die Klasse AVSessionManager übernimmt die Verwaltung von logischen AVSession-Instanzen in dieser Implementierung. Sie bietet Schnittstellen zur GUI und ermöglicht die Steuerung der vorhandenen AVSessions. Es existiert nur eine einzelne Instanz dieser Klasse, die die Befehle der GUI entgegen nimmt und an

die logische AVSession weiterleitet. Der Klasse AVSessionManager kommt eine zentrale Bedeutung zu, da sie die gesamten von UPnP-AV vorgesehenen Fähigkeiten kapselt und so dem Programmierer notwendige Methoden und Events zur Verfügung stellt. Die zentrale Datenstruktur ist eine Hash-Tabelle namens „runningAVSessions“. Innerhalb dieser Datenstruktur werden unter Verwendung eines Keys – in unserem Falle stellt dieser Key ein Renderer-Device-Objekt dar – Werte abgespeichert. Diese Werte wiederum stellen erneut Hash-Tabellen dar, wobei hier die von UPnP-AV verwendeten ConnectionIDs als Key dienen und die zugehörigen Werte AVSession-Instanzen entsprechen. Es ergibt sich somit eine logische Hierarchie, die einen effizienten Zugriff auf die vorhandenen AVSessions ermöglicht. Mit Hilfe eines Renderer-Device-Objektes kann nun die zu diesem Device zugehörige Hash-Tabelle ermittelt werden, die die Informationen über die zur Zeit vorhandenen AVSessions auf diesem speziellen Device enthält. Unter Anwendung der passenden ConnectionID erlangt man so Zugriff auf die logische AVSession, die Schnittstellen enthält, mit deren Hilfe sich diese spezielle AVSession steuern lässt. Diese Klasse erlaubt es, neue AVSessions zu initiieren, aber ermöglicht auch bereits vorhandene Session innerhalb des UPnP-Netzwerks zu steuern. Um eine Trennung zwischen GUI und Funktionalität zu garantieren, wird in der Variablen „activeAVSession“ die Referenz auf die zur Zeit in der GUI steuerbare AVSession gespeichert. So erhält die GUI nur Session-bezogene Informationen zu dieser aktiven Verbindung. Durch einen Session-Switch kann die GUI den AVSessionManager darüber informieren, dass nun eine andere AVSession gesteuert werden soll. Die GUI als solche behält daher allein einen darstellenden Charakter und verfügt über keine tiefer gehende Funktionalität mit Steuerungscharakter. Werden im UPnP-Netzwerk neue Geräte gefunden, so wird der AVSessionManager über dieses neue Device informiert. Er erzeugt nun für dieses Device die speziellen Instanzen, auf die später eingegangen wird, wobei er sich zuvor über vorhandene, offene Sessions informieren muss. Die gesamte Verwaltung offener Verbindungen, sowohl von selbst erstellten als auch von anderen UPnP-konformen Kontrollpunkten initiierten, können an dieser Stelle durchgeführt werden. Durch eine abstrakte Darstellung dieser realen Verbindung durch AVSession-Objekte kann so – logisch gesehen – ein steuernder Zugriff auf die Session erfolgen. Ebenso ist dieses Objekt in der Lage selbständig neue Verbindungen zu bekannten Renderern völlig automatisch aufzubauen, falls der Nutzer für eine spezielle AVSession den Follow-Me-Modus aktiviert. Um einen umfassenden Einblick in die Funktionalität dieser Klasse zu gewähren, sollen nun alle in dieser Klasse vorhandenen Methoden und Events detailliert vorgestellt, beschrieben und kritisch untersucht werden.

Variable: activeAVSession Sichtbarkeit: private

Zugriffsmodifier: static

Typ: AVSession

Diese Variable hält eine Referenz auf die zur Zeit aktive Session, die von der GUI aus vom Benutzer gesteuert werden kann. Es muss sichergestellt werden, dass das Wechselspiel zwischen GUI-Switch und einem automatischen Switch zu keiner inkonsistenten Sicht führt.

Variable: activeThread Sichtbarkeit: private

Typ: Thread

Diese Variable wird verwendet um innerhalb des AVSessionManagers weiterhin Zugriff auf einen eventuell offenen Thread zu haben, der zum aktuellen Zeitpunkt den Zeitfortschritt beim Medien-Abspielvorgang bestimmt. Die Referenz auf diesen Thread, in dem die Methode „RelativePositionTimer“ ausgeführt wird, hält diese Variable.

Variable: avTransports Sichtbarkeit: private

Zugriffsmodifier: static

Typ: Hashtable

Aus Konsistenzgründen wird für diese Variable ebenfalls eine Hashtabelle verwendet. Der Key stellt das UPnP-Device-Objekt dar, mit dessen Hilfe ein Zugriff auf die zu diesem Device gehörige AVTransport-Instanz möglich ist. Da zu jedem Device nur maximal eine AVTransport-Instanz gehören kann, kann die Verwendung einer Hashtabelle als Datenstruktur durchaus in Frage gestellt werden, da es letztendlich nur darauf hinausläuft, ein 2er Tupel aus einer UPnPDevice- und AVTransport-Instanz abzuspeichern.

Variable: connectionManagers Sichtbarkeit: private

Zugriffsmodifier: static

Typ: Hashtable

Aus Konsistenzgründen wird für diese Variable ebenfalls eine Hashtabelle verwendet. Der Key stellt das UPnP-Device-Objekt dar, mit dessen Hilfe ein Zugriff auf die zu diesem Device gehörige ConnectionManager-Instanz möglich ist. Da zu jedem Device nur maximal eine ConnectionManager-Instanz gehören kann, kann die Verwendung einer Hashtabelle als Datenstruktur durchaus in Frage gestellt werden, da es letztendlich nur darauf hinausläuft, ein 2er Tupel aus einer UPnPDevice- und ConnectionManager-Instanz abzuspeichern.

Variable: followMeAVSession Sichtbarkeit: private

Zugriffsmodifier: static

Typ: AVSession

In dieser Variablen wird die Referenz auf die sich zur Zeit im „Follow-Me“-Modus befindliche AVSession-Instanz gespeichert.

Variable: lockedObject Sichtbarkeit: private

Typ: Object

Bei dieser Variablen handelt es sich um eine Hilfskonstruktion zur Thread-Synchronisation.

Variable: renderingControls Sichtbarkeit: private

Zugriffsmodifier: static

Typ: Hashtable

Aus Konsistenzgründen wird für diese Variable ebenfalls eine Hashtabelle verwendet. Der Key stellt das UPnP-Device-Objekt dar, mit dessen Hilfe ein Zugriff auf die zu

diesem Device gehörige RenderingControl-Instanz möglich ist. Da zu jedem Device nur maximal eine RenderingControl-Instanz gehören kann, kann die Verwendung einer Hashtabelle als Datenstruktur durchaus in Frage gestellt werden, da es letztendlich nur darauf hinausläuft, ein 2er Tupel aus einer UPnPDevice- und RenderingControl-Instanz abzuspeichern.

Variable: runningAVSessions Sichtbarkeit: private

Zugriffsmodifier: static

Typ: Hashtable

Diese Variable ist von zentrale Bedeutung innerhalb dieser Klasse. Es handelt sich um zwei verschachtelte Hashtabellen. Zunächst erfolgt unter Verwendung des Keys, der das Renderer-Device-Objekt darstellt, ein Zugriff auf die Hashtabelle, dessen Key wiederum der in UPnP spezifizierten ConnectionID entspricht und dessen zugeordneter Wert nun ein AVSession-Objekt ist. Der Zugriff auf diese Variable muss synchronisiert erfolgen, da dieser von einer Vielzahl von Threads gleichzeitig angefordert werden kann.

Variable: runningAVSessionViaFollowMe Sichtbarkeit: private

Zugriffsmodifier: static

Typ: ArrayList

Diese Variable wird verwendet, um eine Liste von den AVSession-Instanzen zu verwalten, die durch den aktivierten Follow-Me-Modus aufgebaut wurden. Sie ist notwendig, um sicherzustellen, dass bei Erreichen eines Renderers auf dem bereits – aufgrund des Follow-Me-Modus – ein Abspielvorgang initiiert wurde, kein erneuter Vorgang gestartet wird. Es erschien zunächst sinnvoll, dass durch den „Follow-Me-Modus“ weitere Abspielvorgänge initiiert werden, ohne die bereits existierenden zu beenden. Je nach Auslegung des Follow-Me-Modus kann natürlich zu einem späteren Zeitpunkt auch eine andere Sichtweise implementiert werden. Die Daten werden in einer ArrayList-Instanz gespeichert. Um langwierige Iterationsvorgänge zu sparen, könnte auch die gleiche Datenstruktur genutzt werden, die für die Variablen „runningAVSessions“ verwendet wird. Geschachtelte Hash-Tabellen scheinen ein adäquates Design-Prinzip darzustellen. Der Key der ersten Hash-Tabelle stellt das UPnPDevice-Objekt dar mit dessen Hilfe Zugriff auf eine weitere Hash-Tabelle genommen werden kann. Der Key dieser Tabelle stellt die ConnectionID dar und der Wert nun die entsprechende AVSession-Instanz, die auf dem referenzierten Gerät durch den Follow-Me-Modus initiiert wurde.

Variable: sessionsInfoFromServers Sichtbarkeit: private

Zugriffsmodifier: static

Typ: Hashtable

Innerhalb dieser geschachtelten Hash-Tabelle werden AVSession-Instanzen gespeichert, die serverseitig angeboten werden. Wichtig wird diese Datenstruktur, wenn der Server eine „AVTransport“-Instanz anbietet und der Kontrollpunkt ein UPnP-basiertes Netzwerk betritt, wobei der Server für ein aktuell etablierte Verbindung den AVTransport-Dienst anbietet.

Event: OnActiveAVSessionInfo Eingabe: UPnPDevice device, AVSession avSession, CommonStateInfo stateInfo

Ausgabe: void

Dieses Event wird dafür verwendet, die GUI mit sämtlichen zur Verfügung stehenden Informationen über die aktive AVSession zu versorgen. Als Eingabe erhält die Methode das Renderer-Objekt sowie die dazugehörige AVSession-Instanz, auf welche sich die Information in Form eines CommonStateInfo-Objektes bezieht. Das CommonStateInfo-Objekt stellt nichts weiter dar als ein 2er-Tupel, bestehend aus einer Bezeichnung und dessen Wert. Als Beispiel sei genannt: (Volume, 100). Mit Hilfe dieser Informations-Generalisierung ist es möglich, jede Art von Datum an die GUI weiterzuleiten, da die beiden Felder jedes logische Objekt aufnehmen können. Es obliegt daher der GUI diese Information passend aufzubereiten und anzuzeigen. Dieses Event leitet z.B. alle Änderungen, die sich aus der „LastChange“-Benachrichtigung der aktiven AVSession, ergeben an die GUI weiter. Somit kann der Nutzer über sämtliche Änderungen die seine aktiven AVSessions betreffen informiert werden. Um ein besseres Verständnis zu erhalten, sei das folgende Szenario genannt: Eine von diesem Kontrollpunkt initiierte AVSession wird von einem anderen Kontrollpunkt modifiziert. Das könnte z.B. eine Lautstärke-Veränderung oder das Versetzen in den Pause-Zustand sein. Unter Anwendung dieses Events informiert jetzt das AVSessionManager-Objekt die GUI über die Status-Veränderung dieser momentan gesteuert AV-Session. Das AVSessionManager-Objekt nimmt also eine Filterung vor und leitet nur die Information an die GUI weiter, die für den Nutzer momentan relevant sind.

Event: OnActiveAVSessionSwitched Eingabe: AVSession avSession

Ausgabe: void

Dieses Event informiert die GUI darüber, dass sich die aktive AVSession verändert hat. Das geschieht z.B. automatisch, wenn der Nutzer ein gewünschtes Medium auf einem Renderer abspielen möchte. Ist der Verbindu

Event: OnAVSessionCompleted Eingabe: UPnPDevice device, AVSession avSession

Ausgabe: void

Dieses Event informiert darüber, dass der durch „avSession“-abstrahierte Abspielvorgang auf dem Wiedergabegerät „device“ abgebrochen wurde. Es ist zu beachten, dass zwischen dem Stoppen eines Abspielvorgangs und dessen Abbruch ein Unterschied existiert. Wiedergabe-Vorgänge müssen explizit abgebrochen werden.

Event: OnActiveRendererInfo Eingabe: UPnPDevice device, CommonStateInfo stateInfo

Ausgabe: void

Dieses Event informiert die GUI über Veränderungen, die die vorhandenen Renderer betreffen. Wird eine AVSession auf einem Renderer initiiert oder beendet, so wird die GUI über diese Statusänderung unter Anwendung dieses Events unterrichtet.

Event: OnAVSessionInfo Eingabe: UPnPDevice device, AVSession avSession, CommonStateInfo stateInfo

Ausgabe: void

Unter Verwendung dieses Events kann die GUI über Veränderungen, die alle vorhandenen AVSessions betreffen informiert werden. Zum aktuellen Zeitpunkt erfolgt keine Filterung auf Ebene des AVSessionManager-Objektes, also werden alle Veränderungen aller AVSessions an die GUI weitergeleitet. Das Paar UPnPDevice und AVSession beschreibt die Quelle der empfangenen Status-Veränderung, wobei das CommonStateInfo-Objekt wiederum die Information der Änderung enthält. Momentan nimmt die GUI diese Informationen entgegen, um so z.B. die Transport-Stati der AVSessions innerhalb des TreeViews zu aktualisieren. Alle weiteren Informationen werden zur Zeit nicht verarbeitet. Dieser Overhead an Information wird aber bewusst in Kauf genommen, um so eine leichte Erweiterbarkeit zukünftiger Versionen zu garantieren.

Event: OnDebugInfo Eingabe: String text

Ausgabe: void

Diese Event wird verwendet, um die Debug-Konsole zu aktualisieren, wobei die durch „text“ angegebene Meldung dem bereits in der Konsole befindlichen Text vorangestellt wird. Diese Ausgabe soll dem Benutzer die Möglichkeit des Debuggings einräumen, wobei sich die Ausgabe zum jetzigen Zeitraum nur auf den Follow-Me-Modus beschränkt. Es ist aber problemlos möglich, die bereits in der Intel-Implementierung ? vorhandenen Debug-Informationen mit Hilfe dieses Events an die Konsole weiterzuleiten.

Event: OnErrorOccurredOnAVSession Eingabe: AVSession avSession, UPnPInvokeException ex

Ausgabe: void

Tritt innerhalb einer bestehenden AVSession ein Fehler auf, so wird dieses Event verwendet, um den Nutzer über das aufgetretene Problem zu informieren.

Event: OnPausing Eingabe: AVSession avSession, UPnPDevice renderer, CDSObject cdsObject

Ausgabe: void

Dieses Event wird zur Zeit nicht verwendet. An dieser Stelle kann eine Methode angemeldet werden, die aufgerufen wird, wenn der Pause-Zustand der aktuellen AVSession hergestellt wurde. Die AVSession bezeichnet das betroffene AVSession-Objekt, das UPnPDevice den entsprechenden Renderer und das CDSObject enthält – falls vorhanden – das von dem Zustand betroffene Medium.

Event: OnPlaying Eingabe: AVSession avSession, UPnPDevice renderer, CDSObject cdsObject

Ausgabe: void

Dieses Event wird zur Zeit nicht verwendet. An dieser Stelle kann eine Methode angemeldet werden, die aufgerufen wird, wenn der Play-Zustand der aktuellen AVSession

hergestellt wurde. Die AVSession bezeichnet das betroffene AVSession-Objekt, das UPnPDevice den entsprechenden Renderer und das CDSObject enthält – falls vorhanden – das von dem Zustand betroffene Medium.

Event: OnStopped Eingabe: AVSession avSession, UPnPDevice renderer, CDSObject cdsObject

Ausgabe: void

Dieses Event wird zur Zeit nicht verwendet. An dieser Stelle kann eine Methode angemeldet werden, die aufgerufen wird, wenn der Stop-Zustand der aktuellen AVSession hergestellt wurde. Die AVSession bezeichnet das betroffene AVSession-Objekt, das UPnPDevice den entsprechenden Renderer und das CDSObject enthält – falls vorhanden – das von dem Zustand betroffene Medium.

Event: OnUpdateInterface Eingabe: AVSession avSession

Ausgabe: void

Dieses Event wird verwendet, um die GUI darüber zu informieren, dass die aktuelle Ansicht aufgrund eines nicht mehr verfügbaren Renderers aktualisiert werden muss. Der Parameter beinhaltet die nun aktuelle AVSession-Instanz, die sich aufgrund des Verschwindens des Renderers verändert haben kann. Ist von dem Verschwinden des Renderers die aktuelle AVSession-Instanz betroffen, so wird der Player auf einen Standard-Wert zurückgesetzt und die aktive AVSession erhält den Wert „null“.

Event: OnVolumeTo Eingabe: AVSession avSession, UPnPDevice renderer, CDSObject cdsObject

Ausgabe: void

Dieses Event wird zur Zeit nicht verwendet. An dieser Stelle kann eine Methode angemeldet werden, die aufgerufen wird, wenn das Lautstärke-Level der aktuellen AVSession vom Nutzer modifiziert wurde. Die AVSession bezeichnet das betroffene AVSession-Objekt, das UPnPDevice den entsprechenden Renderer und das CDSObject enthält – falls vorhanden – das von der Lautstärke-Änderung betroffene Medium.

Methode: AVSessionManager Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor, mit dessen Hilfe das AVSessionManager-Objekt instantiiert wird. Die vier zentrale Datenstrukturen „connectionManagers“, „av-Transports“, „renderingControls“ und „runningAVSessions“, welche jeweils Hash-Tabellen darstellen, werden ebenfalls erzeugt.

Methode: CancelAVSession Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Durch den Aufruf dieser Methode wird eine bestehende Medien-Wiedergabe vollständig abgebrochen, indem die UPnP-Aktion „ConnectionComplete“ aufgerufen wird.

Methode: CancelAVSessionSink Sichtbarkeit: private

Eingabe: ConnectionManager sender, Int32 ConnectionID, UPnPInvokeException ex, object _Tag

Ausgabe: void

Diese Methode dient als Callback und aktualisiert die internen Datenstruktur. Mit Hilfe der Events „OnUpdateInterface“ und „OnAVSessionCompleted“ wird die GUI über die Modifikation informiert.

Methode: ConnectingToExistingAVSession Sichtbarkeit: private

Eingabe: AVSession avSession, UPnPDevice server, UPnPDevice renderer

Ausgabe: void

Mit Hilfe dieser Methode wird eine logische Verbindung zu einer bereits existierenden AVSession hergestellt. Dabei handelt es sich um eine AVSession, die nicht vom Benutzer selbst erzeugt wurde. Als Eingabe-Parameter verlangt sie die neu hergestellte AVSession in Form einer AVSession-Instanz sowie den Server, von dem das Medien-Objekt stammt und den Renderer, auf dem das Medium zur Zeit wiedergegeben wird. Diese Methode hat nur einen Verwaltungscharakter, d.h. in die „runningAVSession“-Hash-Tabelle wird in die Hash-Tabelle des Renderer-Objektes, die ConnectionID der neuen Verbindung als Key sowie das AVSession-Objekt als Wert eingetragen. Eine Eintragung erfolgt nur, falls die AVSession noch nicht bekannt ist. Gerufen wird diese Methode beim ersten Eintritt in das UPnP-Netzwerk durch die Methode „GetCurrentConnectionIDsRendererSink“, die zu einem späteren Zeitpunkt besprochen wird. Grob zusammengefasst, liefert die Methode „GetCurrentConnectionIDsRendererSink“, die zur Zeit auf einem Renderer aktiven Verbindungen. Jeder Abspieltvorgang wird logisch gesehen durch eine AVSession-Instanz repräsentiert, die fest mit einem Medien-Objekt und dem aktuellen Wiedergabegerät in Form eines UPnPDevice-Objektes verbunden ist.

Methode: DeviceAdded Sichtbarkeit: public

Eingabe: UPnPDevice device

Ausgabe: void

Diese Methode wird niemals direkt gerufen, sondern ist bei dem Event „OnAddedDevice“ in DeviceManager angemeldet, das das AVSessionManager-Objekt darüber informiert, falls ein neuer Server oder Renderer das UPnP-Netzwerk betreten hat. Diese Methode ist ein zentrales Element im AVSessionManager. Für jede Renderer-Instanz werden drei zentrale Steuerungsinstanzen erzeugt. Dabei handelt es sich um Instanzen der Klassen „ConnectionManager“, „AVTransport“ und „RenderingControl“. Ebenso erfolgt eine Subscription für die drei UPnPServices gleichen Namens, wodurch dieser Kontrollpunkt über Änderungen in Form von LastChange-Nachrichten informiert wird. Eine Implementierung auf Server-Seite ist zu ergänzen. Es erweist sich aber als problematisch, innerhalb einer existierenden Verbindung sowohl die Quelle als auch die Senke zu ermitteln. Da

innerhalb dieser Implementierung eine Orientierung am Renderer vorgenommen wurde, fällt es schwer, bei einer bekannten Verbindung von Seiten des Renderer auf den passenden Server zu schließen. Streaming, wie es UPnP vorsieht, ist daher zum aktuellen Zeitpunkt höchstens unter Einschränkungen möglich. Da zu Testzwecken kein adäquater Server zur Verfügung stand, konnte das Verhalten dieser Implementierung in diesem Szenario nicht getestet werden.

Methode: Dispose Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Die Methode wird für ein sauberes Beenden des Programms verwendet. Der Steuerungsthread für den Player wird abgebrochen und die erfolgten Subscriptions werden rückgängig gemacht. Aus Sicherheitsgründen erfolgt auch ein umfassendes Event-Canceling, um sicherzustellen, dass das Programm vollständig terminiert.

Methode: ErrorOccurredOnAVSessionSink Sichtbarkeit: private

Eingabe: AVSession avSession, UPnPInvokeException ex

Ausgabe: void

Sollte es aus irgendwelchen Gründen zu einem Problem auf einer bestehenden AVSession kommen, so wird das dazugehörige AVSession-Objekt aus der „runningAVSessions“ entfernt und per Event „OnErrorOccurredOnAVSession“ wird die GUI über das aufgetretene Problem informiert.

Methode: EstablishNewAVSession Sichtbarkeit: public

Eingabe: UPnPDevice server, UPnPDevice renderer, CDSObject cdsObject

Ausgabe: void

Diese Methode wird als zentrale Anlaufstelle verwendet, wenn vom Nutzer eine neue AVSession initiiert wird. Der Server bezeichnet dabei die Position des Medien-Objektes in Form einer CDSObject-Instanz und der Renderer das gewünschte Abspielgerät. Der korrekte Konstruktor für eine neue AVSession wird gerufen, die dann den Vorgang – wie in UPnP-AV beschrieben – durchführt. Näheres zu dieser Klasse findet sich in dieser Dokumentation. An den Konstruktor werden nun zwei Methodenzeiger übergeben. Dabei wird bei erfolgreichem Aufbau der Verbindung die Methode „NewConnectionEstablishedSink“ aufgerufen und die Methode „ErrorOccurredOnAVSessionSink“ bei Misserfolg.

Methode: EstablishNewAVSessionFollowMe Sichtbarkeit: public

Eingabe: UPnPDevice device, CDSObject cdsObject, String useProtocol, Uri avTransportURI

Ausgabe: void

Diese Methode kann als ein Hilfskonstrukt betrachtet werden, das eine neue AVSession herstellt, falls sich die Position des Nutzers verändert hat. Ist in dessen Nähe ein Renderer, der für das Abspielen des durch den Follow-Me-Modus betroffenen Mediums,

geeignet ist, so erfolgt automatisch der Aufbau einer neuen AVSession. Zum Entwurfszeitpunkt wurde nicht bedacht, dass es nicht trivial möglich ist, das zu einer Session gehörige Medium zu ermitteln. Um die bereits verwendeten Methoden nicht modifizieren zu müssen, erfolgte die Einführung dieser speziellen Lösung. Eine gemeinsame Schnittstelle zur Herstellung einer AVSession wäre an dieser Stelle wünschenswert. Befindet sich eine AVSession im Follow-Me-Modus, die vom User selbst initiiert wurde, so ist auch das dazugehörige Medienobjekt in Form einer CDSObject-Instanz bekannt und es kann genauso verfahren werden wie bei einem neuen AVSession-Aufbau nach einem „Browse“ oder „Search“. In diesem speziellen Fall kann daher nicht auf die Methode „EstablishNewAVSession“ zurückgegriffen werden, da eine Eintragung der AVSession in „runningAVSessionViaFollowMe“ notwendig ist und eine Modifikation der bestehenden Architektur nicht vorgenommen werden sollte.

Methode: EstablishNewAVSessionUseUnknownCDSObject Sichtbarkeit: public

Eingabe: UPnPDevice renderer, String useProtocol, Uri avTransportURI

Ausgabe: void

Da es in der UPnP-Spezifikation nicht ohne weiteres möglich ist, den Server zu bestimmen, von dem ein Medium stammt, das zur Zeit abgespielt wird, wurde diese Hilfsmethode eingeführt. Sie wird nur verwendet, falls eine AVSession in den Follow-Me-Modus versetzt wird, die nicht von diesem Kontrollpunkt initiiert wurde. Es kann natürlich hinterfragt werden, warum ein Trennung dieser beiden Szenarien überhaupt notwendig ist. Das Problem besteht darin, dass maximal das Transferprotokoll des aktuell vom Renderer abgespielten Mediums abgefragt werden kann. Die UPnP-Methode „GetCurrentConnectionInfo“ ? liefert das verwendete Protokoll. Daher kann nun für den automatischen Aufbau einer AVSession nur dieses damals verwendete Protokoll genutzt werden, da Alternativ-Protokolle nicht zur Verfügung stehen, wenn der Server nicht bekannt ist. Erreicht man einen Renderer, der z.B. nur Streaming-Möglichkeiten besitzt, aber keinen Transfer per HTTP unterstützt, so kann dieser das Medium nicht wiedergeben, falls der Renderer auf welchem zur Zeit dasjenige Medium abgespielt werden, dessen zugehörige AVSession in den Follow-Me-Modus versetzt werden soll, bei Abspielbeginn einen Transfer per HTTP durchgeführt hat. Auch wenn der Quell-Server Medien-Streaming unterstützen würde kann von dem abspielenden Renderer nicht trivialerweise auf den Server geschlossen werden. Der Abspielvorgang kann also nicht starten, obwohl dieser theoretisch möglich sein würde. Daher steht nur das zu verwendende Protokoll in Form des Parameters „useProtocol“ zur Verfügung sowie die AVTransportURI, die für die Datenübertragung verwendet wurde, repräsentiert durch den Parameter „avTransportURI“. Das UPnPDevice bezeichnet den Renderer, auf dem die neue AVSession initiiert werden soll.

Methode: GetAllConnectionManagers Sichtbarkeit: public

Zugriffsmodifier: static

Eingabe: void

Ausgabe: Hashtable

Diese Methode liefert eine Hash-Tabelle bestehend aus allen UPnPDevice-Instanzen und den entsprechenden ConnectionManager-Objekten zurück.

Methode: GetAVTransport Sichtbarkeit: public

Zugriffsmodifier: static

Eingabe: UPnPDevice device

Ausgabe: AVTransport

Unter Verwendung dieser Methode kann die zu einem Device gehörige AVTransport-Instanz ermittelt werden. Eine Speicherung innerhalb des UPnPDevice-Objektes wäre dankbar, dennoch sollte die bestehende UPnP-Implementierung von Intel unabhängig von der UPnP-AV-Erweiterung anwendbar bleiben, so dass diese Referenz an einem anderen Ort gespeichert werden musste.

Methode: GetConnectionManager Sichtbarkeit: public

Zugriffsmodifier: static

Eingabe: UPnPDevice device

Ausgabe: AVTransport

Unter Verwendung dieser Methode kann die zu einem Device gehörige ConnectionManager-Instanz ermittelt werden. Eine Speicherung innerhalb des UPnPDevice-Objektes wäre denkbar, dennoch sollte die bestehende UPnP-Implementierung von Intel unabhängig von der UPnP-AV-Erweiterung anwendbar bleiben, so dass diese Referenz an einem anderen Ort gespeichert werden musste.

Methode: GetRenderingControl Sichtbarkeit: public

Zugriffsmodifier: static

Eingabe: UPnPDevice device

Ausgabe: AVTransport

Unter Verwendung dieser Methode kann die zu einem Device gehörige RenderingControl-Instanz ermittelt werden. Eine Speicherung innerhalb des UPnPDevice-Objektes wäre denkbar, dennoch sollte die bestehende UPnP-Implementierung von Intel unabhängig von der UPnP-AV-Erweiterung anwendbar bleiben, so dass diese Referenz an einem anderen Ort gespeichert werden musste.

Methode: GetCurrenAVSessionsInfo Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Diese Methode wird verwendet, um die Informationen zu allen bekannten AVSessions abzufragen. Diese Informationen werden von der GUI angefordert und ausgegeben. Um eine spätere GUI-Modifikation bereits jetzt schon bestmöglich zu unterstützen, erfolgt keine Filterung der Inhalte. Momentan werden diese Informationen verwendet, um den Baum bestehend aus Renderern und AVSessions aufzubauen. Das Event namens „OnAVSessionInfo“ stellt diese Daten bereit. Übergeben wird das Renderer-Objekt, eine aktuell dort laufende AVSession – repräsentiert durch ein AVSession-Objekt – und ein

CommonState-Objekt. Da zum Entwurfszeitpunkt sichergestellt werden sollte, dass die Verzahnung mit der GUI minimal ist, beschreibt das CommonState-Objekt ein 2er-Tupel bestehend aus „(Name, Wert)“.

Methode: GetCurrentConnectionIDsInfo Sichtbarkeit: private

Eingabe: ConnectionManager sender, String value

Ausgabe: void

Diese Methode wird niemals direkt gerufen, sondern wird durch das Event „OnStateVariable.CurrentConnectionIDs“ in der Klasse „ConnectionManager“ aufgerufen. Dieses Event wird ausgelöst, wenn das Device diesen Kontrollpunkt per Eventing über Änderungen innerhalb der offenen Verbindungen informiert. Diese Methode wird ausschließlich dafür verwendet, um die GUI über diese Veränderung zu informieren, um in dem Fall, daß ein Renderer markiert ist, die Modifikation direkt anzeigen zu können.

Methode: GetCurrentConnectionIDsRendererSink Sichtbarkeit: private

Eingabe: ConnectionManager sender, String ConnectionIDs

Ausgabe: void

Wurde per Eventing eine Veränderung der offenen Verbindung propagiert, wird diese Methode verwendet, um die interne Datenstruktur „runningAVSessions“ zu aktualisieren. Ist das Device, das dieses Event ausgelöst hat, noch nicht bekannt, so erfolgt vorab eine Eintragung in „runningAVSessions“, so dass sich alle aktiven Geräte in dieser Datenstruktur befinden. Durch das Event „OnAVSessionInfo“ wird dieses „neue“ Device an die GUI gemeldet und kann ausgegeben werden, auch wenn noch keine Session-Informationen verfügbar sind. Dies kann zwei Gründe haben: Entweder auf dem gewünschten Renderer ist noch keine Session aktiv oder aber die AVSession-Objekte zu den offenen Verbindungen sind noch nicht aufgebaut worden, können aber dann – falls sie aufgebaut wurden – durch das Event „OnAVSessionInfo“ nachgereicht werden. Sollte eine Verbindung nicht mehr aktiv sein, so wird sie aus der Hashtabelle „runningAVSessions“ gelöscht.

Methode: GetCurrentConnectionIDsServerSink Sichtbarkeit: private

Eingabe: ConnectionManager sender, String ConnectionIDs

Ausgabe: void

Diese Methode wurde nachträglich hinzugefügt, um Verbindungen theoretisch unterstützen zu können, die eine AVTransport-Instanz auf Seiten des Servers nutzen. An dieser Stelle wird die Hashtabelle „sessionsInfoFormServers“ verwendet, die einen identischen Aufbau wie die Hashtabelle „runningAVSessions“ besitzt. Der Schlüssel der Hashtabelle „sessionsInfoFormServers“ stellt nun das Server-Objekt dar, mit dessen Hilfe Zugriff genommen werden kann auf die untergeordnete Hashtabelle, die als Schlüssel die ConnectionID und als zugeordneten Wert eine AVSession-Instanz verlangt. Sollte die ConnectionID des Servers noch nicht bekannt sein, so wird sie zunächst eingetragen und mit einem null-Wert versehen, da zum jetzigen Zeitpunkt noch nicht entschieden werden kann, mit welcher AVSession eine Verknüpfung herzustellen ist. Bis zum jetzigen Zeitpunkt

konnte diese Methode nicht getestet werden aufgrund einer fehlenden serverseitigen Unterstützung des UPnP-AVTransport-Dienstes.

Methode: GetSinkProtocolInfo Sichtbarkeit: private

Eingabe: ConnectionManager sender, String value

Ausgabe: void

Sollte eine Modifikation der Sink-Protokolle vorgenommen worden sein, die angeben, mit welchen Protokollen eine Verbindung zu diesem Device hergestellt werden kann, so wird die GUI durch das Event „OnActiveRendererInfo“ über diese Änderung informiert. Der Parameter „sender“ beschreibt das von der Modifikation der Sink-Protokolle betroffene Objekt und der String „value“ enthält Komma-separiert die verfügbaren Protokolle.

Methode: GetSourceProtocolInfo Sichtbarkeit: private

Eingabe: ConnectionManager sender, String value

Ausgabe: void

Sollte eine Modifikation der Source-Protokolle vorgenommen worden sein, die angeben, mit welchen Protokollen ein Medientransfer von diesem Device aus durchgeführt werden kann, so wird die GUI durch das Event „OnActiveRendererInfo“ über diese Änderung informiert. Der Parameter „sender“ beschreibt das von der Modifikation der Source-Protokolle betroffene Objekt und der String „value“ enthält Komma-separiert die neuen Protokolle. Sollte es sich um ein reines Renderer-Device handeln, so besitzt dieses keine Source-Protokolle, sondern nur Sink-Protokolle.

Methode: LastChangesAVTransport Sichtbarkeit: private

Eingabe: AVTransport avTransport, CommonStateInfo stateInfo

Ausgabe: void

Diese Methode ist von zentraler Bedeutung innerhalb der vorliegenden Implementierung. Sie nimmt Änderungen entgegen, die von Renderern oder Servern per „LastChange“ propagiert wurden, und erzeugt per Event „OnAVSessionInfo“ passende Ereignisse für die GUI. Gehört die Modifikation zu der aktiven AVSession, so wird zusätzlich das Event „OnActiveAVSession“ geworfen, um ein GUI-Update bzgl. der zur Zeit steuerbaren AVSession durchführen zu können. Ebenso wird sichergestellt, dass wenn der neue Status der aktiven Verbindung den Wert „PLAYING“ annimmt, auch die aktuelle Abspiel-Position ermittelt wird. Sollte der Status der Verbindung den Wert „STOPPED“ erreichen, so wird der Timer, der den Medienfortschritt beschreibt, unverzüglich beendet.

Methode: LastChangesRenderingControl Sichtbarkeit: private

Eingabe: RenderingControl renderingControl, CommonStateInfo stateInfo

Ausgabe: void

Diese Methode ist ebenfalls von zentraler Bedeutung innerhalb der vorliegenden Implementierung. Sie nimmt Änderungen, die von Renderern per „LastChange“ propagiert wurden entgegen und erzeugt per Event „OnAVSessionInfo“ passende Ereignisse für

die GUI. Gehört die Modifikation zu der aktiven AVSession, so wird zusätzlich das Event „OnActiveAVSession“ geworfen, um ein GUI-Update bzgl. der zur Zeit steuerbaren AVSession durchführen zu können.

Methode: NearestCompatibleRenderer Sichtbarkeit: public

Eingabe: UPnPDevice device, CDSObject cdsObject, String useProtocol, Uri avTransportURI

Ausgabe: void

Diese Methode wird durch das Event „OnNearestCompatibleRenderer“ im DeviceManager gerufen. Zunächst muss getestet werden, ob auf dem übergebenen Renderer eine neue Medien-Wiedergabe initiiert werden muss. An dieser Stelle wird die ArrayList-Instanz „runningAVSessionViaFollowMe“ verwendet, welche die durch den Follow-Me-Modus initiierten AVSession-Instanzen enthält. Mit Hilfe einer Listen-Iteration wird überprüft, ob bereits auf diesem Renderer schon eine Follow-Me-basierte Verbindung etabliert wurde. Ist dies der Fall, so wird keine neue Verbindung initiiert, ansonsten wird durch den Aufruf der Methode „EstablishNewAVSessionFollowMe“ eine neue Medien-Wiedergabe gestartet.

Methode: NewAVSessionInfoReceived Sichtbarkeit: private

Eingabe: AVTransport avTransport, CommonStateInfo stateInfo

Ausgabe: void

Von der Implementierung betrachtet, unterscheidet sich diese Methode nur wenig von der bereits besprochenen Methode namens „LastChangesAVTransport“. Von der logischen Sichtweise betrachtet, ist aber ein Unterschied zu erkennen, so dass bewusst eine weitere Methode eingeführt wurde, um diese logische Trennung zu unterstreichen. Die Methode „LastChangesAVTransport“ wird nur von außen durch LastChange-Nachrichten angestoßen, während diese Methode aktiv gerufen wird. Passive Informationen in Form von LastChange-Nachrichten erreichen die Methode „LastChangesAVTransport“, während eine aktive Informationsanforderung in Form von aktiven Informationen wie z. B. durch den Aufruf der UPnP-Aktion „GetPositionInfo“ nur diese Methode passieren. Da zu späteren Zeitpunkten möglicherweise eine solche Unterscheidung sinnvoll sein kann, bestand die Veranlassung, diese Methode einzuführen.

Methode: NewConnectionEstablishedSink Sichtbarkeit: private

Eingabe: AVSession avSession, UPnPDevice server, UPnPDevice renderer, Uri currentURI, CDSObject cdsObject, bool toPlay

Ausgabe: void

Wurde eine neue Nutzer initiierte Verbindung erfolgreich aufgebaut, so wird diese Methode als Callback aufgerufen. Erst zu diesem Zeitpunkt erfolgt eine Eintragung in „runningAVSessions“ und ein aktiver AVSession-Switch. Der Parameter „avSession“ beschreibt die neue logische Verbindung, das Server- sowie das Renderer-Objekt stellen die beiden Endpunkte der Verbindung dar, die URI beschreibt die AVTransportURI, die CDSObject-Instanz liefert eine Referenz auf das Medienobjekt und der boolesche

Parameter „toPlay“ liefert die Information, ob zusätzlich die durch UPnP-spezifizierte Aktion „Play“ zu rufen ist. Dieser Aufruf ist jedoch nur dann notwendig, wenn der vorherige Abspielstatus des Renderers noch nicht dem Status „PLAYING“ entsprach.

Methode: Pause Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Die Methode wird verwendet, wenn der Abspielvorgang der aktuellen AVSession auf „PAUSED_PLAYBACK“ gesetzt werden soll. Natürlich macht ein Aufruf dieser Methode nur dann Sinn, wenn zuvor eine zu steuernde AVSession-Instanz ausgewählt worden ist.

Methode: PauseSink Sichtbarkeit: private

Eingabe: AVSession avSession, UPnPDevice renderer, CDSObject cdsObject

Ausgabe: void

Diese Methode stellt einen Callback dar, welcher aufgerufen wird, wenn der Abspielvorgang der zu steuernden AVSession pausiert. Die Parameter avSession, renderer und cdsObject bezeichnen die von der Statusänderung betroffenen Objekte.

Methode: Play Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Die Methode wird verwendet, wenn der Abspielvorgang der aktuellen AVSession auf „PLAYING“ gesetzt werden soll. Natürlich macht ein Aufruf dieser Methode nur dann Sinn, wenn zuvor eine zu steuernde AVSession ausgewählt worden ist.

Methode: PlaySink Sichtbarkeit: private

Eingabe: AVSession avSession, UPnPDevice renderer, CDSObject cdsObject

Ausgabe: void

Diese Methode stellt einen Callback dar, welcher aufgerufen wird, wenn der Abspielvorgang der zu steuernden AVSession den Status „PLAYING“ angenommen hat.

Methode: RelativeTimePositionTimer Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Diese Methode wird in einem Thread durch den Aufruf der Methode „StartRelativeTimePositionTimer“ gestartet. Sie hat die Aufgabe, den Zeitfortschritt seit Abspielbeginn eines Mediums darzustellen und die GUI darüber zu informieren. Dieser Thread ist fest mit der aktuellen AVSession verbunden. Ein AVSession-Switch führt zu einem Threadabbruch und zu einem Neustart. Da das Netz nicht unnötigerweise belastet werden soll wird nur alle 5 Sekunden mit Hilfe der Methode „GetPositionInfo“ eine Anfrage nach dem Zeitfortschritt innerhalb des Mediums gestellt. Währenddessen läuft die eigene relative Zeit ungehindert weiter. Da unter allen Umständen vermieden werden musste, dass in

die eigentliche UPnP-Status-Variable geschrieben wird, musste ein recht umständlicher Umweg gewählt werden. Dieser Umweg wurde notwendig, da ansonsten nicht sichergestellt werden konnte, dass Modifikationen durch andere Kontrollpunkte erkannt werden können. Die Idee bestand darin, dass wenn die Abweichung der Zeit in der UPnP-Status-Variable „RelativeTimePosition“ echt größer oder echt kleiner als die Summe aus dem bekannten Wert dieser Variablen bei Timer-Beginn und der bereits fortgeschriebenen Zeit ist, so muss es sich um eine Verschiebung handeln, die berücksichtigt werden muss. Diese Verschiebung könnte dadurch entstanden sein, indem ein anderer Nutzer einen Vor- oder Rückspulvorgang gestartet hat. Tritt nun eine echte Differenz ein, so wird der eigentliche Timer zurückgesetzt und mit der nun aktualisierten Zeit fortgefahren. Natürlich kann man auf diese Weise nur eine Näherung erreichen, da z.B. Verzögerungszeiten beim Nachrichtentransfer nicht berücksichtigt wurden. Dies ließe sich durch eine Mitteilung der Nachrichtenlaufzeiten erreichen. Dennoch scheint der Nutzen dieser Vorgehensweise fraglich, so dass darauf bewusst verzichtet wurde. Ein Thread-Abbruch mit Hilfe eines Flags führt zu massiven Problemen bei schnellen AVSession-Switches, welche unter einer Sekunde durchgeführt werden, da ansonsten nicht sichergestellt werden kann, dass sich der Zeitfortschritt auch tatsächlich auf die korrekte Session bezieht. Es musste sichergestellt werden, dass der Thread unverzüglich und ohne weitere Wartezeit beendet wird.

Methode: RemovedRenderer Sichtbarkeit: public

Eingabe: UPnPDevice device

Ausgabe: void

Mit Hilfe dieser Methode wird ein Renderer-Objekt sowie alle zu dem Device gehörigen AVSessions aus „runningAVSession“ gelöscht. Sollte die zur Zeit steuerbare AVSession davon betroffen sein, so wird mit Hilfe des Events „OnUpdateInterface“ die GUI über diesen Vorfall unterrichtet. An dieser Stelle wird – falls betroffen – die Hilfsdatenstruktur namens „runningAVSessionViaFollowMe“ aktualisiert. In [1.1.1](#) wird näher auf diese Datenstruktur eingegangen.

Methode: RemovedServer Sichtbarkeit: public

Eingabe: UPnPDevice device

Ausgabe: void

Da die UPnP-Spezifikation vorsieht, dass auch ein Server z. B. im Falle eines Media-Streamings die AVTransport-Steuerungsinstanz zur Verfügung stellen kann, muss diesem Anwendungsfall Rechnung getragen werden. Steht ein Server innerhalb des UPnP-Netzwerkes aufgrund eines Timer-Ablaufs oder einer „BYE-BYE“-Notifikation nicht mehr zur Verfügung, so muss auch eine eventuell vorhandene AVSession, die eine AVTransport-Instanz auf Server-Seite verwendet, geschlossen werden. Verwendet wird dazu die Datenstruktur namens „sessionsInfoFormServers“, die einen ähnlichen Aufbau wie „runningAVSessions“ aufweist. Ein Key, sprich das Server-Device-Objekt, wird verwendet, um eine Hashtabelle zu erhalten, dessen Key wiederum die ConnectionID darstellt und der dazugehörige Wert einer potentiellen AVSession zugeordnet ist. Um

Inkonsistenzen zum realen UPnP-Netzwerk zu vermeiden, muss diese Datenstruktur aktualisiert werden, um nicht mehr steuerbare AVSessions auszublenden.

Methode: SeekSink Sichtbarkeit: private

Eingabe: AVSession avSession, UPnPDevice renderer, String target

Ausgabe: void

Diese Methode wird bei einem erfolgreich durchgeführten Seek-Vorgang aufgerufen, damit es zu keiner weiteren automatischen Seek-Aktionen auf der betroffenen AVSession-Instanz kommt. Durch das Setzen des „seeked“-Flags kann verhindert werden, dass falls ein Nutzer eine sich im Follow-Me-Modus befindliche AVSession-Instanz anhält und zu einem späteren Zeitpunkt fortsetzt, kein automatischer Seek-Vorgang mehr gestartet wird.

Methode: SetVolumeTo Sichtbarkeit: public

Eingabe: Int32 volume

Ausgabe: void

Diese Methode kann aufgerufen werden, um die Lautstärke der zur Zeit steuerbaren AVSession zu verändern. Der Übergabe-Parameter beschreibt das neue Lautstärke-Level.

Methode: SetVolumeToSink Sichtbarkeit: private

Eingabe: AVSession avSession, UPnPDevice renderer, CDSObject cdsObject

Ausgabe: void

Diese Methode stellt einen Callback dar, welcher aufgerufen wird, wenn die Lautstärke der betroffenen „avSession“ auf dem Wiedergabe-Gerät „renderer“ modifiziert wurde. Der Parameter „cdsObject“ bezeichnet – falls bekannt – das von der Lautstärke-Änderung betroffene Medien-Objekt.

Methode: StartRelativeTimePositionTimer Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Diese Methode startet einen Thread mit dessen Hilfe das Fortschreiten der Zeit innerhalb des Players in der GUI dargestellt wird. Aus Effizienzgründen wird nur ein einzelner Thread verwendet. Erfolgt innerhalb der GUI ein AVSession-Switch, so wird – falls vorhanden – der bisherige Thread per „Abort()“ unterbrochen und ein neuer gestartet. Die dadurch erzeugte „ThreadAbortException“ wird in der Methode „RelativeTimePositionTimer“ abgefangen, muss aber nicht weiter behandelt werden.

Methode: Stop Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Die Methode wird verwendet, wenn der Abspielvorgang der aktuellen AVSession auf „STOPPED“ gesetzt werden soll. Natürlich macht ein Aufruf dieser Methode nur dann Sinn, wenn zuvor eine zu steuernde AVSession ausgewählt worden ist.

Methode: StopSink Sichtbarkeit: private

Eingabe: AVSession avSession, UPnPDevice renderer, CDSObject cdsObject

Ausgabe: void

Diese Methode stellt einen Callback dar, der aufgerufen wird, wenn der Abspielvorgang der zu steuernden AVSession den Status „STOPPED“ eingenommen hat.

Methode: StopRelativeTimePositionTimer Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Diese Methode wird verwendet um den Thread, in dem die Methode „RelativePositionTimer“ ausgeführt wird, zu beenden.

Methode: SwitchAVSession Sichtbarkeit: public

Eingabe: AVSession avSession

Ausgabe: void

Diese Methode hat die Aufgabe einen AVSession-Switch durchzuführen, falls z. B. in der GUI eine andere Session zur Steuerung markiert wurde. Diese Methode wird auf zwei Arten verwendet. Da in der GUI durch die Knoten-Selektierung im verwendeten TreeView automatisch diese Methode gerufen wird, müssen zwei Fälle separat betrachtet werden: (1) Der markierte Knoten entspricht nicht der aktiven Session: Es wird ein echter Switch durchgeführt, indem die GUI per Event „OnActiveAVSessionSwitched“ über den Switch informiert wird. (2) Der markierte Knoten entspricht der aktiven Session: Da bei einem Haupt-Menüpunkt-Wechsel die gesamten Session-Informationen neu aufgebaut werden und die aktive Session im Baum markiert wird, löst dies in der GUI ein „AfterSelect“-Event aus. Da es sich dabei aber nicht um einen echten Switch handelt, werden durch den Aufruf dieser Methode nur die zu dieser Session gehörigen Informationen per Event an die GUI gemeldet. Verwendet wird an dieser Stelle das Event namens „OnActiveAVSessionInfo“. Der Parameter „avSession“ bezeichnet die AVSession, auf die umgeschaltet werden soll. Im Falle der Gleichheit kann es sich nicht um einen „echten“ Switch handeln, so dass der GUI allein die bekannten Daten zu dieser Session gemeldet werden.

Property: FollowMeAVSession Sichtbarkeit: public

Eingabe: AVSession

Ausgabe: AVSession

Mit Hilfe dieser Property kann Zugriff genommen werden auf die AVSession-Instanz, die sich momentan im Follow-Me-Modus befindet.

Klasse: ContentManager Ähnlich zu der Klasse „AVSessionManager“ erfüllt auch ein Objekt dieser Klasse komplexe Management-Aufgaben innerhalb der vorliegenden Implementierung. Mit seiner Hilfe werden alle Aufgaben, die in Verbindung mit der Medien-Navigation stehen, gesteuert. Dies umfasst z. B. „Browse“- oder „Search“-Anfragen. Im

Gegensatz zu dem AVSessionManager, wo es darum geht, steuernd auf bestehende Verbindungen einzugehen oder neue Verbindung zu initiieren, geht es im ContentManager, um die Aufgabenerfüllung bezüglich aller zu einem Medium vorhandenen Informationen. Er übernimmt die Verwaltung der vom User initiierten ContentSession. In der Klasse ContentSession wird detailliert erläutert, welche abstrakte Aufgabe eine „ContentSession“ innerhalb dieser Implementierung erfüllt. Sie soll eine Schnittstelle zwischen GUI und den jeweiligen ContentSessions bieten, und komplexe Aufgaben wie Suche und Sortierung übernehmen. Weiterhin werden die Ergebnisse für die GUI passend aufbereiten und an die zuständigen Stellen für die Ausgabe weitergeben. Wichtig war es, die Ausgabe von der inneren Logik sauber zu trennen. Zentrale Datenstrukturen sind „relationCDSObjectListContentSession“, sowie die „CDSObjectList“-Instanz „activeCDSObjectList“. Die Grundidee ist die folgende: Die ArrayList-Instanz „relationCDSObjectListContentSession“ speichert 2er Tupel, bestehend aus einer „ContentSession“-Instanz sowie einem „CDSObjectList“-Objekt. Die „CDSObjectList“-Instanz enthält ihrerseits eine ArrayList-Instanz, die Medien-Objekte oder Container aufnehmen kann. Wie bereits angesprochen, werden – wie in der UPnP-Spezifikation beschrieben – Medienobjekte als „CDSObject“-Instanzen aufgefasst. Diese Klassenhierarchie wird an späterer Stelle in diesem Dokument beschrieben. Die „CDSObjectList“-Instanz speichert eine sortierte Reihenfolge der Medienobjekte, die an die GUI geliefert werden können. Um ein intuitiveres Verständnis dieser Architektur zu erlangen, hilft es möglicherweise, sich eine „CDSObjectList“-Instanz als eine logische GUI-Ansicht vorzustellen. Eine CDSObjectList-Instanz, die in der GUI ausgegeben wird, besitzt in der dazugehörigen „CDSObjectList“-Instanz die gleiche Position. Da aber die in der GUI dargestellten Medienobjekte nicht zwangsweise von dem selben Server stammen müssen, besteht die Idee darin, mit Hilfe einer „ContentSession“-Instanz eine Medienanfrage, wie beispielsweise ein „Browse“ oder „Search“ logisch zu abstrahieren und als Instanz einer solchen Klasse darzustellen. Wird nun eine Server-übergreifende Suche durchgeführt, z.B. nach allen im UPnP-Netzwerk verfügbaren Audio-Objekten, so müssen an allen Servern, die eine Such-Aktion implementieren, Anfragen geschickt werden. In diesem Fall entspricht eine Suchanfrage auf einem Server genau einer Content-Session-Instanz. Daher ist es möglich, dass eine CDSObjectList-Instanz mit mehreren Content-Session-Objekten assoziiert ist. Dadurch wird erreicht, dass die einer CDSObjectList-Instanz zugehörigen Suchanfragen bzw. ContentSession-Objekte, passend eingefügt werden, um so dem Nutzer eine vollständig Speicherort-transparente Medienliste präsentieren zu können. Im Zentrum steht also das Medium und nicht sein Speicherort. Wird eine Suchanfrage der eine „Browse“-Operation auf nur einem Server durchgeführt, so ist die CDSObjectList-Instanz auch nur mit einem ContentSession-Objekt verbunden. Die Variable „relationCDSObjectListContentSession“ wird verwendet um die Zugehörigkeit einer ContentSession zu einer speziellen CDSObjectList-Instanz als Repräsentant einer Mediensammlung zu speichern. Die Variable „activeCDSObjectList“ als Instanz der Klasse CDSObjectList entspricht der logischen Medienansicht innerhalb der GUI. Die in ihr enthaltenden Medien-Objekte werden – falls ausgewählt – in der GUI angezeigt. Jede GUI-Ansicht, die sich auf die Medien-Ausgabe bezieht, beinhaltet daher im Hintergrund stets eine CDSObjectList-Instanz. Die weiteren CDSObjectList-Instanzen „activeCDSObjectListFiltered“ und „activeCD-

SObjectListNotFiltered" beinhalten jeweils gefilterte logische Medienansichten. Je nach Wahl ist entweder die „activeCDSObjectList“ gleich der „activeCDSObjectListFiltered“ oder bei deaktiviertem Medienfilter gleich „activeCDSObjectListNotFiltered“. Es blieb jedoch das Problem zu lösen, wie die GUI über die erhaltenen Medienobjekte informiert wird. Im Prinzip gibt es nur zwei Varianten: Entweder werden die Medienobjekte vollständig geparkt und dann in einer Listenstruktur an die GUI weitergegeben oder aber jedes Medienobjekt wird direkt an die GUI gereicht, sobald es vollständig in eine Objektstruktur zerlegt wurde. In dieser Implementierung entschieden sich die Entwickler für die 2. Variante. Da der Parse-Vorgang recht zeitaufwendig ist, kann so eine Performance-Steigerung erreicht werden und der Nutzer kann frühzeitig über Ergebnisse informiert werden. Unter Angabe eines Index kann die GUI das Objekt an die passende Position zeichnen und benötigt daher keine weitere „Intelligenz“, da die gesamten Medien-Operationen wie z. B. eine Sortierung allein von einer unteren Schicht, dem „ContentManager“, übernommen werden.

Zusammengefasst bedeutet dies nun folgendes: Wird eine DIDL-Nachricht von der ContentSession-Instanz empfangen, so zerlegt diese die Medienobjekte aus der XML-Struktur in eine logische Objektstruktur. Diese Objektstruktur wird in 1.1.2 erläutert. Sobald ein Medienobjekt vollständig geparkt wurde, also eine CDSObject-Instanz entstanden ist, erfolgt die Übergabe dieses Objektes an den „ContentManager“. Dieser hat nun die Aufgabe, das Objekt bzgl. des gewünschten Sortierkriteriums in die Medienliste, also der aktuellen „CDSObjectList“-Instanz einzusortieren. Ist dies geschehen, so wird die GUI per Event „OnSingleCDSObjectSorted“ darüber informiert und erhält als Eingabe-Parameter die CDSObject-Instanz, die Index-Position, an die das Objekt zu zeichnen ist, die entsprechende guiID sowie das GUI-Element, auf das dieses Objekt gezeichnet werden muss.

An einigen Stellen mag es aufgefallen sein, dass eine GUI-Referenz in diese Klasse Einzug gehalten hat. Die Erläuterung der Begründung dafür ist nicht ganz einfach, dennoch soll ambitionierten Entwicklern das dadurch gelöste Problem beschrieben werden, damit zukünftig Fehler vermieden werden können. Das bereits angesprochene Prinzip, dass geparkte Medienobjekte jeweils einzeln an die GUI per Event gemeldet werden, erwies sich als problematisch. Erfolgte ein Ansichtswechsel durch den Benutzer, bevor alle Medienobjekte per Event an die GUI gemeldet werden konnten, so musste sichergestellt werden, dass die noch folgenden Objekte nicht mehr gezeichnet werden. Natürlich war es möglich, die noch folgenden Events abubrechen, indem „activeCDSObjectList“ einfach auf null gesetzt wurde, aber die sich bereits auf dem Weg zur GUI befindlichen Elemente konnten nicht mehr abgefangen werden. Auch die Modifikation der guiID erfolgte zu spät, so dass immer noch Objekte an die GUI gemeldet wurden. Bei Entwicklungsbeginn sollte versucht werden für die Medienausgabe mit nur einem ListView-Objekt auszukommen, das je nach Ansicht eine Modifikation erfährt. Dies erwies sich durch das angesprochenen Problem als nicht länger haltbar. Durch den Ansichtswechsel konnte zwar das ListView-Objekt aktualisiert und modifiziert werden, aber Medienobjekte wurden auch nach dem Switch noch in diese Liste gezeichnet und ausgegeben. Die Idee, dass nur ein Switch möglich ist, wenn sich keine Medienobjekte mehr in der „Pipeline“ zur GUI-Ausgabe befinden, erwies sich ohne eine „Semaphore“, die im Compact .NET

Framework nicht existiert, als nicht durchsetzbar, da ein Synchronisationselement verwendet werden musste, das auch von einem anderen Thread modifiziert werden kann. Der Versuch, eine „Semaphore“ mit Hilfe zweier Mutex-Elemente und eines synchronisierten Increments „nachzubauen“, endete in einigen Situationen in einem Deadlock. Um nun eine funktionierende Version erreichen zu können, bestand die Idee darin, bei einem Ansichtswechsel ein neues ListView-Objekt zu erzeugen und das vorherige in den Hintergrund zu versetzen. Durch Angabe des spezifischen ListView-Objekt gelang es, das angesprochene Problem zu vermeiden. Die Control-Instanz, die einige Methoden als Parameter entgegen nimmt, hält eine Referenz auf das GUI-Objekt, in das später, die von der ContentSession zurückgelieferten Anfrageergebnisse, gezeichnet werden sollen. „Veraltete“ Objekte innerhalb der Pipeline werden somit in das dann nicht mehr sichtbare ListView-Objekt gezeichnet. Es ist jedoch wichtig zu verstehen, dass diese Methoden nur durch die Control-Instanz durchgetunnelt werden, um sie später an die GUI weiterzugeben. Es erfolgt kein direkter GUI-Zugriff oder eine Ausgabe-Modifikation. Daher konnte diese Instanz bewusst allgemein gehalten werden, da die meisten GUI-Komponenten in C# von der Klasse „Control“ erben, kann hier für eine spätere Modifikationen ein beliebiges GUI-Element angegeben werden.

Durch den Abbruch unter Verwendung von „CancelAllEvents“ gelangen nur sehr wenige Elemente in die Pipeline bis der ContentManager weitere, per Event gemeldete Medienobjekte, stoppen kann, da kein gültiges „activecdsobjectList“-Objekt vorliegt. Gleichzeitig erfolgt eine Abmeldung von den Events „OnSingleCDSObjectParsed“ und „OnResponseReceived“ der betroffenen ContentSession-Instanzen.

Variable: activeCDSObjectList Sichtbarkeit: private

Zugriffsmodifier: static

Typ: CDSObjectList

Diese Variable hält eine Referenz auf die aktuelle Medienliste vom Typ CDSObjectList. Die Belegung dieser Variable ist im Zusammenhang mit den Variablen „activeCDSObjectListFiltered“ und „activeCDSObjectListNotFiltered“ zu sehen. Die gespeicherte Referenz verweist entweder auf „activeCDSObjectListFiltered“ oder „activeCDSObjectListNotFiltered“ je nach dem, welche Liste in der GUI angezeigt werden soll.

Variable: activeCDSObjectListFiltered Sichtbarkeit: private

Zugriffsmodifier: static

Typ: CDSObjectList

Diese Variable hält eine Referenz auf die aktuelle Medienliste vom Typ CDSObjectList, die nur Medien enthält, die von dem aktuell ausgewählten Renderer aufgrund eines positiven Protokoll-Matchings abgespielt werden können. Die gespeicherte Liste ist dynamischen Abläufen unterworfen und wird je nach Renderer-Auswahl in der GUI neu erstellt.

Variable: activeCDSObjectListNotFiltered Sichtbarkeit: private

Zugriffsmodifier: static

Typ: CDSObjectList

Diese Variable hält eine Referenz auf die aktuelle ungefilterte Medienliste vom Typ CDSObjectList.

Variable: contentDirectoryOnServers Sichtbarkeit: private

Zugriffsmodifier: static

Typ: ArrayList

Die Variable speichert in einer ArrayList alle zur Verfügung stehenden ContentDirectory-Instanzen für alle verfügbaren Server.

Variable: numberReturned Sichtbarkeit: private

Typ: Int32

Diese Variable nimmt die Anzahl der eingegangenen Medien-Objekte aus dem Event „OnResponseReceived“ entgegen. Wird z. B. eine server-weite Suche durchgeführt, so speichert sowie die Summe der Medien-Objekte aus den eingegangenen Such-Antworten.

Variable: numberReturnedFiltered Sichtbarkeit: private

Typ: Int32

Ähnlich wie „numberReturned“ arbeitet auch diese Variable. Sie enthält die Summe der zum ausgewählten Medien-Renderer kompatiblen Medien-Objekte.

Variable: relationCDSObjectListContentSession Sichtbarkeit: private

Zugriffsmodifier: static

Typ: ArrayList

In dieser ArrayList werden Structs (2er Tupel) gespeichert, die eine Verknüpfung zwischen ContentSession-Instanz und zugehöriger CDSObjectList-Instanz darstellen. Auf diese Weise kann die übergeordnete Medienliste referenziert werden.

Variable: selectedRenderer Sichtbarkeit: private

Zugriffsmodifier: static

Typ: UPnPDevice

Diese Variable hält eine Referenz auf den zur Zeit ausgewählten Renderer in Form einer UPnPDevice-Instanz. Die Entscheidung diese Information im DeviceManager unterzubringen, wurde getroffen da sie auch außerhalb des Content-Managers verfügbar sein muss. Im Sinne der entworfenen Architektur sollte diese Information jedoch ausgelagert werden. Eine zukünftige Version könnte dies korrigieren.

Variable: toFilterOnSelectedRenderer Sichtbarkeit: private

Zugriffsmodifier: static

Typ: bool

Mit Hilfe dieses Flags kann eine Entscheidung getroffen werden, ob der präsentierte Inhalt automatisch zu filtern ist, d.h. auf Kompatibilität zu dem zur Zeit ausgewählten Renderer zu prüfen ist.

Variable: toLock Sichtbarkeit: private

Typ: Object

Bei dieser Variablen handelt es sich um eine Hilfskonstruktion zur Thread-Synchronisation.

Event: OnErrorOccurred Eingabe: UPnPInvokeException ex

Ausgabe: void

Tritt ein allgemeiner Fehler ein, so wird dieses Event verwendet, um den Nutzer über das aufgetretene Problem zu informieren.

Event: OnErrorOccurredOnContentSession Eingabe: ContentSession contentSession, UPnPInvokeException ex

Ausgabe: void

Tritt innerhalb einer bestehenden ContentSession-Instanz ein Fehler auf, so wird dieses Event verwendet, um den Nutzer über das aufgetretene Problem zu informieren.

Event: OnMetadataReceived Eingabe: CDSObject cdsObject

Ausgabe: void

Die durch eine CDSObject-Instanz repräsentierten Metadaten werden mit Hilfe dieses Event weitergeleitet. Dieses Event wird zum jetzigen Zeitpunkt nicht verwendet.

Event: OnResponseReceived Eingabe: ContentSession contentSession, Int32 number-Returned

Ausgabe: void

Dieses Event meldet den Eingang einer serverseitigen Antwort auf eine Such- oder Browse-Anfrage. Übergeben wird die betroffene ContentSession-Instanz, sowie die Anzahl passender Medien-Objekte. Mit Hilfe dieses Event kann der Nutzer frühzeitig über die Anzahl der zu seiner Anfrage passenden Medienobjekte informiert werden, auch wenn der Parse-Vorgang der Medien-Objekte noch nicht abgeschlossen ist.

Event: OnSearchCapabilites Eingabe: UPnPDevice device, String[] searchCapabilites

Ausgabe: void

Dieses Event meldet den Eingang der angeforderten „searchCapabilites“ vom Server. Die Komma-separierte Liste wird in ein String-Array überführt und unter Angabe des dazugehörigen Gerätes weitergeleitet.

Event: OnSingleCDSObjectSorted Eingabe: CDSObject cdsObject, int index, int gui-ID, Control control

Ausgabe: void

Dieses Event ist von zentraler Bedeutung. Es meldet an die GUI ein vollständig geparses Medien-Objekt mit einem Index, der die Position angibt, an der es einzufügen ist. Der Index ergibt sich aus dem gewünschten Sortierkriterium. Auf diese Weise erhält die GUI maximale Schnelligkeit bei der Medien-Präsentation. Auch wenn die Medienliste

noch nicht vollständig aufgebaut wurde, können die bereits gefundenen und geparsten Medien-Objekte weitergeleitet werden.

Event: OnSortCapabilites Eingabe: UPnPDevice device, String[] sortCapabilites

Ausgabe: void

Dieses Event meldet den Eingang der angeforderten „sortCapabilites“ vom Server. Die Komma-separierte Liste wird in ein String-Array überführt und unter Angabe des zugehörigen Gerätes weitergeleitet.

Event: OnSystemUpdateID Eingabe: UPnPDevice device, UInt32 systemUpdateID

Ausgabe: void

Dieses Event meldet den Eingang der angeforderten „systemUpdateID“ vom Server. Diese Information wird in der aktuellen Implementierung nicht verwendet, erlaubt aber ein späteres Cachen bereits besuchter Inhalte. Da diese ID nur im Falle einer Änderung am Server neu gesetzt wird, ist es möglich – im Falle seltener Änderungen – ein komplexes Caching-Verfahren einzusetzen, das die Effizienz beim Browse-Vorgang enorm verbessern kann.

Methode: AllCDSObjectsParsedSink Sichtbarkeit: private

Eingabe: ContentSession contentSession

Ausgabe: void

Wurden alle zu einer ContentSession gehörigen CDSObjekt-Instanz erfolgreich in eine Objektstruktur zerlegt, so kann aus der Datenstruktur „relationCDSObjectListContentSession“, diese spezifische ContentSession herausgenommen werden, da keine weiteren Resultate mehr produziert werden können. Eine ContentSession stellt daher eine abstrakte Verbindung zwischen einer Content-Anfrage und einer serverseitigen Antwort dar. Diese Methode wird als Callback gerufen, und zwar genau dann, wenn die ContentSession die von einem Server erhaltende DIDL-Nachricht erfolgreich und vollständig geparst hat. Durch das Event „OnAllCDSObjectParsed“ im ContentSession-Objekt erfolgt ein Aufruf dieser Methode, wobei die ContentSession-Instanz übergeben wird, die ihre Arbeit vollständig beendet hat. Da diese Methode mehrfach von verschiedenen Threads aus gerufen werden kann, ist eine Synchronisation notwendig, die mit Hilfe eines „Lock“ sichergestellt wurde.

Methode: BrowseDirectChildren Sichtbarkeit: public

Eingabe: UPnPDevice device, CDSContainer cdsContainer, String sortCriteria, String filter, UInt32 startingIndex, UInt32 requestedCount, int guiID, Control control

Ausgabe: void

Diese Methode erlaubt es, innerhalb einer vom Server bereitgestellten Medien-Struktur, bestehend aus Medien-Archiven und atomaren Medien, zu browsen. Durch ihren Aufruf wird mit Hilfe der Methode „EstablishContentSession“ eine neue ContentSession-Instanz erzeugt. Unter Verwendung dieser ContentSession kann nun mittels „BrowseDirectChildren“ die UPnP-AV definierte „Browse“-Aktion aufgerufen werden. Die Anzahl

der zurückgelieferten Medienobjekte wird jeweils mit 0 initialisiert. Es erfolgt eine Anmeldung am Event „OnResponseReceived“ der ContentSession-Instanz, um so vor dem Abschluss des Parse-Vorgangs die Anzahl der vom Server zurückgelieferten Medienobjekte zu erhalten. Registriert wird dort die Methode „NumberReturned“, die später durch das angesprochene Event aufgerufen wird. Als Eingabe nimmt diese Methode die UPnPDevice-Instanz des Servers, das CDSContainer-Objekt, dessen Inhalt angefordert werden soll, sowie Sortier- und Filterkriterien in Form des Strings „sortCriteria“ bzw. „filter“ entgegen. Des Weiteren kann der Start-Index und die Anzahl der gewünschten Medienobjekte angegeben werden. In dieser Implementierung sind jedoch Start-Index und Anzahl der gewünschten maximalen Resultate stets 0, wodurch alle in dem CDS-Container enthaltenen Objekte angefordert werden. Leider ließ es sich nicht vermeiden, zwei GUI-spezifischen Elemente tief in diese Implementierung einzuschleusen. In diesem Fall handelt es sich um eine „guiID“ sowie eine Control-Instanz. In der Beschreibung der Klasse „ContentManager“ erfolgt eine Erklärung dieses Workarounds.

Methode: BrowseMetadata Sichtbarkeit: public

Eingabe: UPnPDevice device, CDSObject cdsObject, int guiID, Control control

Ausgabe: void

Diese Methode wird in dieser Implementierung bisher nicht verwendet und wurde noch nicht getestet. Ihre Funktionsweise ist aber der Methode „BrowseDirectChildren“ sehr ähnlich. Die Methode „BrowseMetadataSink“ wird als Callback gerufen, wenn die Metadaten des betroffenen Objektes eingetroffen und geparkt worden sind. Als Resultat steht wieder eine „CDSObject“-Instanz zur Verfügung, die per Event „OnMetadataReceived“ an die GUI gemeldet werden kann.

Methode: BrowseMetadataSink Sichtbarkeit: private

Eingabe: ContentSession contentSession, CDSObject cdsObject

Ausgabe: void

Diese Methode dient als Callback nach dem Aufruf der Methode „BrowseMetadata“. Diese Methode liefert die angeforderten Meta-Informationen in Form einer „CDSObject“-Instanz an die GUI weiter. Diese Methode wird in dieser Implementierung bisher nicht verwendet und wurde noch nicht getestet.

Methode: CancelActiveCDSObjectList Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Mit Hilfe dieser Methode werden die Assoziationen zwischen „activeCDSObjectList“ und den dazugehörigen „ContentSession“-Instanzen aufgelöst. Durch das Setzen der Variablen „activeCDSObjectList“, „activeCDSObjectListFiltered“ und „activeCDSObjectListNotFiltered“ auf null, werden noch folgende Events darin gehindert, an die GUI weitergegeben zu werden, da ein Ansichtswechsel erfolgte. Sind alle Objekte innerhalb der ContentSession-Instanz geparkt worden, so erfolgt die automatische Löschung der Assoziation zwischen betroffener ContentSession und dem entsprechenden „activeCDS-

ObjectList". An dieser Stelle könnte eine Optimierung ansetzen, um den im Hintergrund fortgesetzten Parse-Vorgang zu beenden, da die Ergebnisse dieses Vorgangs nicht mehr benötigt werden.

Methode: ContentManager Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Hierbei handelt es sich um den Konstruktor der Klasse „ContentManager“. Dieser initialisiert die Variablen „contentDirectoryOnServers“, „relationCDSObjectListContentSession“, „activeCDSObjectList“, „activeCDSObjectListFiltered“ und „activeCDSObjectListNotFiltered“. In der Variablen „contentDirectoryOnServers“ werden die „ContentDirectory“-Instanzen gespeichert, mit deren Hilfe ein Zugriff auf die UPnP-AV-Aktion ermöglicht wird.

Methode: DeviceAdded Sichtbarkeit: public

Eingabe: UPnPDevice device

Ausgabe: void

Diese Methode wird von dem Event „OnAddedDevice“ im DeviceManager aufgerufen. Wenn ein neuer UPnP-AV-Server hinzugefügt wurde, so wird automatisch eine „ContentDirectory“-Instanz erzeugt, mit deren Hilfe ein Zugriff auf die UPnP-AV-Aktion ermöglicht wird. Diese Referenz wird nun in der ArrayList-Instanz „contentDirectoryOnServers“ zur weiteren Verwendung gespeichert. Des Weiteren werden umgehend die „SearchCapabilites“, die „SortCapabilites“ und die „SystemUpdateID“ abgefragt. Zusätzlich wäre es möglich auch diese Variablen per Subscription anzufordern, um so stets eine aktuelle Sicht der Server-Fähigkeiten zu erhalten. Auf dieses Feature wurde jedoch an dieser Stelle verzichtet, da es recht unwahrscheinlich ist, dass die „SearchCapabilites“ und „SortCapabilites“ einer hohen Modifikationsfrequenz unterworfen sind. Die „SystemUpdateID“ wird beim aktuellen Implementierungsstand nicht benötigt.

Methode: DeviceRemoved Sichtbarkeit: public

Eingabe: UPnPDevice device

Ausgabe: void

- noch zu implementieren -

Methode: ErrorOccurredOnContentSessionSink Sichtbarkeit: private

Eingabe: ContentSession contentSession, UPnPInvokeException ex

Ausgabe: void

Sollte innerhalb einer bestehenden ContentSession ein Fehler entstehen, falls z. B. der Server, auf dem gerade eine „Browse“-Aktion ausgeführt wird, nicht mehr verfügbar sein sollte, so wird diese Methode aufgerufen. Per Event „OnErrorOccurredOnContentSession“ werden die ContentSession-Instanz, in der der Fehler aufgetreten ist, sowie das entsprechende Fehler-Objekt an die GUI weitergeleitet. Zuvor wird jedoch die Assoziation zwischen „CDSObjectList“ und „ContentSession“ aufgelöst, indem das Paar aus

„relationCDSObjectListContentSession“ gelöscht wird, ebenso erfolgt eine Abmeldung von den Events „OnSingleCDSObjectParsed“ und „OnResponseReceived“ in der entsprechenden ContentSession-Instanz.

Methode: EstablishContentSession Sichtbarkeit: private

Eingabe: CDSObjectList cdsObjectList, CDSObject parentCDSObject, UPnPDevice device, String sortCriteria, String filter, ContentSession.SingleCDSObjectParsed callbackSingleCDSObjectParsed, ContentSession.AllCDSObjectsParsed callbackAllCDSObjectsParsed, ContentSession.ErrorOccurred callbackErrorOccurred, int guiID, Control control

Ausgabe: ContentSession contentSession

Diese Methode stellt eine neue ContentSession-Instanz bereit. Es erfolgt eine Eintragung der Assoziation bestehend aus „cdsObjectList“ und „contentSession“ in „relationCDSObjectListContentSession“. Sollte der Parameter „cdsObjectList“ den Übergabewert aufweisen, so wird eine neue „CDSObjectList“-Instanz erzeugt und die Verknüpfung, wie bereits beschrieben, eingetragen. Der Parameter „parentCDSObject“ gibt an, ob eine ContentSession-Instanz an ein bestimmtes Medien-Objekt gebunden ist dessen Inhalt abgefragt werden soll. Dieser Parameter ist nur bei einer „Browse“-Aktion ungleich null. Im Falle einer Suche ist natürlich das Ergebnis nicht an eine bestimmte Medien-Instanz gebunden. Die „UPnPDevice“-Instanz bezeichnet den Server, auf dem eine virtuelle ContentSession etabliert werden soll. Die Strings „sortCriteria“ und „filter“ entsprechen den in UPnP-AV-spezifizierten Belegungen. Den Callbacks kommt eine besondere Bedeutung zu. Die Methode, auf die durch „callbackSingleCDSObjectParsed“ verwiesen wird, wird aufgerufen, sobald ein Medienobjekt aus der Server-Antwort erfolgreich geparsed wurde, in diesem Falle wird die Methode „SingleCDSObjectParsedSink“ im ContentManager aufgerufen, die eine Index-Ermittlung vornimmt, sowie eine potentielle Filterung. Details werden bei den Beschreibungen der Methoden „SingleCDSObjectParsedSink“ im ContentManager und „PutInList“ als Methode einer „CDSObjectList“-Instanz genannt.

Methode: FilterCurrentContent Sichtbarkeit: public

Eingabe: int guiID, Control listView

Ausgabe: void

Diese Methode ist von ihrer Arbeitsweise her recht einfach zu verstehen. Im Zusammenhang mit dem Programmablauf ist es jedoch recht schwierig, die Implementierung zu überblicken. Hauptaufgabe der Methode ist zunächst einmal den aktuell angezeigten Inhalt, genauer gesagt, die Medien-Objekte in „activeCDSObjectList“ zu filtern, indem alle zu einem vorab ausgewählten Renderer inkompatiblen Medien-Objekte aussortiert werden. Die Überprüfung erfolgt mit Hilfe eines Abgleichs zwischen den Transferprotokollen auf Renderer- und Serverseite. Existiert auf Seiten des Renderers ein Transferprotokoll, mit dessen Hilfe es möglich ist, das Medium zu übertragen, so wird es in die Medienliste „activeCDSObjectListFiltered“ aufgenommen. Je nachdem ob der Filter aktiviert oder deaktiviert Filter ist enthält „activeCDSObjectList“ entweder eine Referenz auf die Objekte „activeCDSObjectListFiltered“ oder „activeCDSObjectList-

NotFiltered". Die Methode selbst kann von der GUI gerufen werden. Die Inhalte werden auf Kompatibilität überprüft und im positiven Fall per Event „OnSingleCDSObjectSorted“ an die GUI weitergeleitet. Wird erneut diese Methode gerufen, so erfolgt jeweils ein Switch zwischen der nicht-gefilterten Ansicht und der gefilterten, wobei die gefilterte Ansicht neu erstellt wird. Selbstverständlich kann sich ein Test auf Kompatibilität nur auf atomare Medienobjekte und nicht auf Container beziehen da diese nicht abgepielt werden können. Zentrale Methode ist die bereits erwähnte, aber noch nicht näher erläuterte Methode „PutInList“ der entsprechenden „CDSObjectList“-Instanz. Vorab sei erwähnt, dass diese Methode mit Hilfe einer binären Suche das Medienobjekt nach dem gewünschten Sortierkriterium in die Medienliste einfügt und einen Index zurück liefert, der der Position in dieser Liste entspricht. Die Anzahl der kompatiblen Objekte wird um 1 erhöht und es erfolgt ein GUI-Event.

Sollte eine nicht gefilterte Sicht gewünscht sein, werden alle zu dem Zeitpunkt in der Datenstruktur „SortedCDSObjects“ befindlichen Medien-Objekte per Event „OnSingleCDSObjectSorted“ an die GUI gemeldet. Die Datenstruktur „SortedCDSObjects“ stellt eine ArrayList-Instanz dar und befindet sich innerhalb des Objektes „activeCDSObjectListNotFiltered“. Die separate Meldung eines geparsten Medienobjektes in der vorliegenden Implementierung erweist sich jedoch hier als problematisch. Falls der Filter aktiviert wird, obwohl noch Medienobjekte an die GUI gemeldet werden müssen, liefert diese Methode nur die zum Zeitpunkt der Aktivierung bereits in die Liste „SortedCDSObjects“ eingetragenen Medien-Objekte. Die noch zu parsenden Objekte würden verloren gehen. Dies ist jedoch nicht tolerierbar. An dieser Stelle hilft die Synchronisation der beiden nebenläufigen Methoden „FilterCurrentContent“ und „SingleCDSObjectParsedSink“ mit Hilfe des Sperr-Objektes „toLock“. Kommt nun die Methode „FilterCurrentContent“ zum Zuge, so kann keine nebenläufige Operation in „SingleCDSObjectParsedSink“ fortfahren. Es müssen zunächst alle bereits vorhandenen Medienobjekte an die GUI gemeldet worden sein, bis das Lock wieder freigegeben wird. Durch die nun aktivierte Filterung in „SingleCDSObjectParsedSink“ werden nur die Medienobjekte zur GUI geleitet, die den Filter passiert haben. Die GUI wird also mit Hilfe zweier Operationen aufgebaut. Die Methode „FilterCurrentContent“ liefert per Event die bereits eingetragenen Objekte, während die Methode „SingleCDSObjectParsedSink“ die noch nicht in die Liste „SortedCDSObjects“ eingetragenen Medien-Objekte liefert, die zu den gewählten Renderer kompatibel sind. Die Problematik wird aber im Detail noch einmal in der Methode „SingleCDSObjectParsedSink“ beschrieben.

Methode: GetAllAudioFiles Sichtbarkeit: public

Eingabe: int guiID, Control control

Ausgabe: void

Diese Methode wird verwendet, um alle verfügbaren Medien-Objekte zu erhalten, die vom Typ „object.item.audioItem.musicTrack“ sind. Da für jeden Server ein ContentDirectory-Objekt vorhanden ist, wird nun auf jedem Server eine Suche gestartet, falls diese unterstützt wird. Implementiert ein Server die in UPnP-AV-spezifizierte Suchaktion nicht, so können die dort verfügbaren Inhalte nicht ausgelesen werden. Als

Erweiterung könnte ein rekursives Browsen durch die Verzeichnisstruktur durchgeführt werden, jedoch ist diese Operation enorm zeitaufwendig und wurde daher zum aktuellen Implementierungszeitpunkt nicht programmiert.

An dieser Stelle wird deutlich, dass nun mehrere ContentSession-Instanzen mit einer „CDSObjectList“-Instanz assoziiert sind. Es ist natürlich klar, dass ein Wunsch nach einer Sortierung zwar lokal auf jedem Server möglich sein kann, aber natürlich das Gesamtergebnis noch nicht sortiert ist. Implementieren also n Server die UPnP-AV-Aktion „Search“, so treffen auch n Antwortnachrichten ein, die zwar lokal sortiert sein können, aber nun noch zu einer gesamt-sortierten Liste überführt werden müssen. Dieses Problem wird bei der Beschreibung der Methode „SingleCDSObjectParsedSink“ näher erläutert. Die Eingabeparameter bezeichnen die GUI-Komponenten, auf der die Ausgabe präsentiert werden soll.

Methode: GetAllPictures Sichtbarkeit: public

Eingabe: int guiID, Control control

Ausgabe: void

Diese Methode wird verwendet, um alle verfügbaren Medien-Objekte zu erhalten, die vom Typ „object.item.imageItem“ sind. Da für jeden Server ein ContentDirectory-Objekt vorhanden ist, wird nun auf jedem Server eine Suche gestartet falls diese unterstützt wird. Implementiert ein Server die in UPnP-AV-spezifizierte Such-Aktion nicht, so können die dort verfügbaren Inhalte nicht ausgelesen werden. Als Erweiterung könnte ein rekursives Browsen durch die Verzeichnisstruktur durchgeführt werden, jedoch ist diese Operation enorm zeitaufwendig und wurde daher zum aktuellen Implementierungszeitpunkt nicht programmiert.

An dieser Stelle wird deutlich, dass nun mehrere ContentSession-Instanzen mit einer „CDSObjectList“-Instanz assoziiert sind. Es ist natürlich klar, dass ein Wunsch nach einer Sortierung zwar lokal auf jedem Server möglich sein kann, aber natürlich das Gesamtergebnis noch nicht sortiert ist. Implementieren also n Server die UPnP-AV-Aktion „Search“, so treffen auch n Antwortnachrichten ein, die zwar lokal sortiert sein können, aber nun noch zu einer gesamt-sortierten Liste überführt werden müssen. Dieses Problem wird bei der Beschreibung der Methode „SingleCDSObjectParsedSink“ näher erläutert. Die Eingabeparameter bezeichnen die GUI-Komponenten, auf der die Ausgabe präsentiert werden soll.

Methode: GetAllVideoFiles Sichtbarkeit: public

Eingabe: int guiID, Control control

Ausgabe: void

Diese Methode wird verwendet, um alle verfügbaren Medien-Objekte zu erhalten, die vom Typ „object.item.videoItem“ sind. Da für jeden Server ein ContentDirectory-Objekt vorhanden ist, wird nun auf jedem Server eine Suche gestartet falls diese unterstützt wird. Implementiert ein Server die in UPnP-AV-spezifizierte Such-Aktion nicht, so können die dort verfügbaren Inhalte nicht ausgelesen werden. Als Erweiterung könnte ein rekursives Browsen durch die Verzeichnisstruktur durchgeführt werden, jedoch ist diese Operation

enorm zeitaufwendig und wurde daher zum aktuellen Implementierungszeitpunkt nicht programmiert.

An dieser Stelle wird deutlich, dass nun mehrere ContentSession-Instanzen mit einer „CDSObjectList“-Instanz assoziiert sind. Es ist natürlich klar, dass ein Wunsch nach einer Sortierung zwar lokal auf jedem Server möglich sein kann, aber natürlich das Gesamtergebnis noch nicht sortiert ist. Implementieren also n Server die UPnP-AV-Aktion „Search“, so treffen auch n Antwortnachrichten ein, die zwar lokal sortiert sein können, aber nun noch zu einer gesamt-sortierten Liste überführt werden müssen. Dieses Problem wird bei der Beschreibung der Methode „SingleCDSObjectParsedSink“ näher erläutert. Die Eingabeparameter bezeichnen die GUI-Komponenten, auf der die Ausgabe präsentiert werden soll.

Methode: GetContentDirectory Sichtbarkeit: public

Zugriffsmodifier: static

Eingabe: UPnPDevice device

Ausgabe: ContentDirectory

Diese Methode erlaubt es die zu einem Device gehörigen „ContentDirectory“-Instanz zu ermitteln. Ist eine solche Instanz nicht verfügbar, so liefert diese Methode null zurück.

Methode: GetSearchCapabilites Sichtbarkeit: private

Eingabe: ContentDirectoryStruct contentDirectoryInfo

Ausgabe: void

Die Ausführung dieser Methode führt zu einer Anforderung der „SearchCapabilites“ eines Servers, der durch den Eingabeparameter „contentDirectoryInfo“ angegeben wird.

Methode: GetSearchCapabilitesSink Sichtbarkeit: private

Eingabe: ContentDirectory sender, String SearchCaps, UPnPInvokeException ex, object _Tag

Ausgabe: void

Der Aufruf dieser Methode erfolgt aus der UPnP-AV-Implementierung heraus und übergibt die von der UPnP-AV-Aktion „GetSearchCapabilites“ ermittelten Ergebnisse im Parameter „SearchCaps“. Es erfolgt die Eintragung des Ergebnisses in das entsprechende Struct namens „contentDirectoryInfo“, das mit Hilfe des Parameters „-Tag“ getunnelt wurde.

Methode: GetSearchResults Sichtbarkeit: public

Eingabe: int guiID, Control control, string strTitle, string strCreator, string strType, UPnPDevice device

Ausgabe: void

Diese Methode stellt die Möglichkeit zur Verfügung, eine individuelle Suche auf einem oder mehreren Servern durchzuführen.

Methode: GetSortCapabilites Sichtbarkeit: private

Eingabe: ContentDirectoryStruct contentDirectoryInfo

Ausgabe: void

Die Ausführung dieser Methode führt zu einer Anforderung der „SortCapabilites“ eines Servers, der durch den Eingabeparameter „contentDirectoryInfo“ angegeben wird.

Methode: GetSortCapabilitesSink Sichtbarkeit: private

Eingabe: ContentDirectory sender, String SortCaps, UPnPInvokeException ex, object _Tag

Ausgabe: void

Der Aufruf dieser Methode erfolgt aus der UPnP-AV-Implementierung heraus und übergibt die von der UPnP-AV-Aktion „GetSortCapabilites“ ermittelten Ergebnisse im Parameter „SortCaps“. Es erfolgt die Eintragung des Ergebnisses in das entsprechende Struct namens „contentDirectoryInfo“, das mit Hilfe des Parameters „_Tag“ getunnelt wurde.

Methode: GetSystemUpdateID Sichtbarkeit: private

Eingabe: ContentDirectoryStruct contentDirectoryInfo

Ausgabe: void

Die Ausführung dieser Methode führt zu einer Anforderung der „SystemUpdateID“ eines Servers, der durch den Eingabeparameter „contentDirectoryInfo“ angegeben wird.

Methode: GetSystemUpdateIDSink Sichtbarkeit: private

Eingabe: ContentDirectory sender, UInt32 Id, UPnPInvokeException ex, object _Tag

Ausgabe: void

Der Aufruf dieser Methode erfolgt aus der UPnP-AV-Implementierung heraus und übergibt die von der UPnP-AV-Aktion „GetSystemUpdateID“ ermittelten Ergebnisse im Parameter „Id“. Es erfolgt die Eintragung des Ergebnisses in das entsprechende Struct namens „contentDirectoryInfo“, das mit Hilfe des Parameters „_Tag“ getunnelt wurde.

Methode: IsCompatibleToRenderer Sichtbarkeit: public

Eingabe: UPnPDevice device, CDSObject cdsObject

Ausgabe: bool

Diese Methode wird dafür verwendet, zu überprüfen, ob das angegebene Medien-Objekt zu der übergebenen Renderer-Instanz kompatibel ist. Es wird anhand eines regulären Ausdrucks getestet, ob mindestens ein kompatibles Transferprotokoll verfügbar ist mit dessen Hilfe es möglich ist, den durch das Medium repräsentierten Server, mit dem angegebenen Renderer zu verbinden. Der Rückgabewert gibt an, ob das durch „cdsObject“ repräsentierte Medium, auf dem Renderer abgespielt werden kann oder nicht.

Methode: IsCompatibleToRenderer Sichtbarkeit: public

Zugriffsmodifier: static

Eingabe: UPnPDevice device, String possibleProtocolsNearestRendererInfo, String protocolInfoActiveRenderer

Ausgabe: bool

Diese Methode wurde nachträglich hinzugefügt, um den Follow-Me-Modus unterstützen zu können. Es soll die Entscheidung getroffen werden, ob das für eine damalige Übertragung des Mediums vom Server zum Renderer verwendete Transferprotokoll zu einem der durch „possibleProtocolsNearestRendererInfo“ angegebenen Komma-separierten Protokolle des aktuell am nächsten befindlichen Renderers kompatibel ist. Der Parameter „device“ beschreibt die UPnPDevice-Instanz des Renderers mit der kürzesten Entfernung zum Nutzer. Der Rückgabewert entspricht nun dem Fall, ob das durch „protocolInfoActiveRenderer“ repräsentierte Protokoll für eine Medienübertragung verwendet werden kann. Es wäre von Vorteil, wenn es gelingen würde, diese und die oben genannte Methode aufgrund gleicher Funktionalität zu einer Methode zu verschmelzen.

Methode: NumberReturned Sichtbarkeit: public

Eingabe: ContentSession contentSession, Int32 numberReturned

Ausgabe: void

Diese Methode ist von untergeordneter Bedeutung. Sie wird gerufen, sobald eine Antwort-Nachricht einer „Browse“- oder „Search“-Anfrage vom Server empfangen wurde. Übergeben wird die ContentSession-Instanz sowie die Anzahl der empfangenen Medienobjekte. Da eine Suche – wie bereits angesprochen – sich über mehrere Server erstrecken kann, wird die lokale Variable „numberReturned“ im ContentManager verwendet, um die Gesamtanzahl der zurückgelieferten Medien-Objekten ermitteln zu können. Per Event „OnResponseReceived“ wird ein Event an die GUI geworfen, das vorab den Nutzer über die Anzahl der erhaltenen Medienobjekte informiert.

Methode: RemoveAllEntriesForInactiveCDSObjectList Sichtbarkeit: private

Eingabe: CDSObjectList cdsObjectList

Ausgabe: void

Wurde festgestellt, dass noch geparste Medienobjekte einer vorherigen GUI-Ansicht z. B. aufgrund eines frühzeitigen Switches innerhalb der GUI, vorhanden sind, so wird „relationCDSObjectListContentSession“ aktualisiert, indem alle, zu diesem Objekt gehörigen, noch vorhandenen Einträge vollständig gelöscht werden. Dadurch kann „relationCDSObjectListContentSession“ auf einem aktuellen Stand gehalten werden.

Methode: ServerRemoved Sichtbarkeit: public

Eingabe: UPnPDevice device

Ausgabe: void

- noch zu implementieren -

Methode: SingleCDSObjectParsedSink Sichtbarkeit: private

Eingabe: ContentSession contentSession, CDSObject cdsObject

Ausgabe: void

Diese Methode ist das zentrale Element innerhalb des ContentManagers. Wurde innerhalb der ContentSession ein Medienobjekt in Form einer CDSObject-Instanz erzeugt,

wird dieses unmittelbar nach der Erstellung an den ContentManager weitergeleitet. Dies geschieht, indem innerhalb der ContentSession das Event „OnSingleCDSObjectParsed“ geworfen wird, was zu einem Aufruf dieser Methode führt. Da möglicherweise vor der eigentlichen Ausgabe in der GUI noch eine Sortierung notwendig sein kann, muss der korrekte Index ermittelt werden um die gepasste CDSObject-Instanz an der korrekten Position in der GUI auszugeben. Ist für das neu eingetroffene Objekt zusätzlich eine Filterung durchzuführen, so übernimmt die Methode dies ebenfalls. Es existieren zwei zentrale Variablen, zum einen „activeCDSObjectListNotFiltered“ und zum anderen „activeCDSObjectListFiltered“. Eine neu erzeugte CDSObject-Instanz wird in „activeCDSObjectListNotFiltered“ eingetragen. Innerhalb dieser Variablen werden die Rohdaten gespeichert. Die Datenstruktur „activeCDSObjectListFiltered“ nimmt nur diejenigen Objekte auf, die die Überprüfung auf Kompatibilität mit dem entsprechenden Renderer positiv durchlaufen haben. Da zwischenzeitlich ein Renderer-Switch durchgeführt worden sein kann, muss diese Datenstruktur aktualisiert werden, indem die Überprüfung unter Verwendung des aktuell ausgewählten Renderers durchgeführt wird. Die Datenstruktur „activeCDSObjectListFiltered“ befindet sich so stets auf dem neuesten Stand. Je nach dem, ob eine Filterung erwünscht ist, hält die Variable „activeCDSObjectList“ die Referenz auf eines der beiden Objekte „activeCDSObjectListFiltered“ oder „activeCDSObjectListNotFiltered“. Durch den Aufruf der Methode „PutInList“ wird das Medienobjekt, je nach Angabe des Sortierkriteriums, in der ArrayList-Instanz „SortedCDSObjects“ abgespeichert und an der korrekten Index-Position eingefügt. Eine Einfüge-Operation in die ArrayList-Instanz „activeCDSObjectListNotFiltered“ erfolgt stets, um eine Filter-Deaktivierung unterstützen zu können, ohne dass die Informationen erneut eingeholt werden. Für zukünftige Erweiterungen kann daher die „activeCDSObjectListNotFiltered“ beliebig manipuliert werden. Die Rohdaten stehen für die aktuelle Ansicht weiterhin unverfälscht zur Verfügung.

Klasse: ContentSession Diese Klasse lässt sich im Gegensatz zur AVSession nur recht schwierig einordnen, wenn es um die logische objektorientierte Abstraktion der Realität geht. Zunächst sei erwähnt, dass der ContentManager eine Vielzahl von ContentSession-Instanzen verwalten und koordinieren muss. Eine ContentSession beschreibt die logische Server-Anfrage sowie dessen Antwort. Es gilt also, dass eine ContentSession-Instanz fest mit einem einzelnen Server verbunden ist. Mit ihrer Hilfe wird eine Anfrage bzgl. verfügbarer Medien-Inhalte an den betroffenen Server gestellt. Sie nimmt die Antwort in Form einer in UPnP definierten DIDL-Nachricht entgegen und erzeugt aus dieser XML-Struktur eine Objektstruktur. Mit dem vollständigen Beenden des Parse-Vorgangs endet die Lebensdauer einer solchen ContentSession. Für weitergehende Anfragen werden neue Instanzen erzeugt. Zusammengefasst stellt die ContentSession-Instanz Möglichkeiten bereit, um direkt mit der UPnP-AV-Implementierung kommunizieren zu können. Ähnlich wie in einer Client-Server-Architektur, kapselt dieses Objekt das „Request“ an den Server sowie dessen „Reply“. Nach Beendigung dieses Wechselspiels endet die Lebensdauer dieses Objekts. Es wurde besonderes Augenmerk auf die Performance der einzelnen Methoden gelegt, so dass eine recht effiziente Implementierung entstanden ist, die aber

durchaus noch Optimierungsspielraum lässt. Um ein umfassendes Verständnis dieser Klasse zu ermöglichen, werden nun alle Methoden und Events im Detail vorgestellt und besprochen.

Variable: cdsTypesContainers Sichtbarkeit: private

Typ: ArrayList

Diese Variable stellt eine ArrayListe dar, die die Type-Instanzen aller - im Sinne von UPnP - Container-Klassen enthält.

Variable: cdsTypesItems Sichtbarkeit: private

Typ: ArrayList

Diese Variable stellt eine ArrayListe dar, die die Type-Instanzen aller - im Sinne von UPnP - Item-Klassen enthält.

Variable: contentDirectory Sichtbarkeit: private

Typ: ContentDirectory

Da eine ContentSession fest mit einem Server verbunden ist, speichert diese Variable, die zum diesem Server gehörige Referenz seines contentDirectory-Objekts.

Variable: device Sichtbarkeit: private

Typ: UPnPDevice

Diese Variable speichert die Referenz auf den Server in Form einer UPnPDevice-Instanz.

Variable: filter Sichtbarkeit: private

Typ: String

Diese Variable speichert - im Sinne von UPnP - einen Komma-separierten String mit dem zu verwendenden Filter.

Variable: numberReturned Sichtbarkeit: private

Typ: UInt32

An dieser Stelle wird die Anzahl der zurückgelieferten Medien-Objekte gespeichert.

Variable: searchCriteria Sichtbarkeit: private

Typ: String

Diese Variable speichert - im Sinne von UPnP - einen Komma-separierten String mit dem zu verwendenden Suchkriterium.

Variable: sortCriteria Sichtbarkeit: private

Typ: String

Diese Variable speichert - im Sinne von UPnP - einen Komma-separierten String mit dem zu verwendenden Sortierkriterium.

Variable: totalMatches Sichtbarkeit: private

Typ: UInt32

Diese Variable speichert - im Sinne von UPnP - die Anzahl der Matches.

Variable: updateID Sichtbarkeit: private

Typ: UInt32

Diese Variable speichert die UpdateID.

Event: OnAllCDSObjectParsed Eingabe: ContentSession contentSession

Ausgabe: void

Diese Event informiert darüber, dass alle Medien-Objekte in der zugehörigen ContentSession-Instanz parsed und verarbeitet wurden.

Event: OnErrorOccurred Eingabe: ContentSession contentSession, UPnPInvokeException ex

Ausgabe: void

Tritt in einer bestehenden ContentSession ein Fehler auf, so informiert dieses Event den Nutzer.

Event: OnResponseReceived Eingabe: ContentSession contentSession, Int32 numberReturned

Ausgabe: void

Dieses Event informiert die GUI, über die Anzahl auf der angegebenen ContentSession-Instanz eingegangenen Medien-Objekte.

Event: OnSingleCDSObjectParsed Eingabe: ContentSession contentSession, CDSObject cdsObject

Ausgabe: void

Dieses zentrale Event informiert den ContentManger über ein vollständig geparstes Medien-Objekt. Angemeldet ist derzeit nur die Methode „SingleCDSObjectParsedSink“ im Content-Manager, der die Weiterverarbeitung des eingegangenen Medien-Objekte übernimmt, indem eine Index-Position innerhalb der übergeordneten Liste ermittelt wird.

Methode: BrowseDirectChildren Sichtbarkeit: public

Eingabe: CDSContainer cdsContainer, UInt32 startingIndex, UInt32 requestedCount

Ausgabe: void

Mit Hilfe dieser Methode wird eine Anforderung des Inhalts eines durch den Parameter „cdsContainer“ angegebenen Containers durchgeführt. Unter Verwendung der entsprechenden „ContentDirectory“-Instanz wird so direkt mit der UPnP AV-Implementierung kommuniziert.

Methode: BrowseDirectChildrenSink Sichtbarkeit: private

Eingabe: ContentDirectory sender, String ObjectID, ContentDirectory.Enum_A_ARG_TYPE_BrowseFlag BrowseFlag, String Filter, UInt32 StartingIndex, UInt32 RequestedCount, String SortCriteria, String Result, UInt32 NumberReturned, UInt32 TotalMatches, UInt32 UpdateID, UPnPInvokeException ex, object _Tag

Ausgabe: void

Diese Methode nimmt die serverseitige Antwort zu einer „BrowseDirectChildren“-Anfrage entgegen und wird von der UPnP-AV-Implementierung nach Durchführung der Anfrage gerufen. Der String „ObjectID“ bezeichnet die ID des Containers, dessen Inhalt als XML-konformer String in der Variablen „Result“ übergeben wird. Diese DIDL-Nachricht bedarf nun einer weiteren Aufbereitung, um die im DIDL-Format beschriebenen Medien in eine objektorientierte Struktur zu überführen. Die weiteren Parameter können in der UPnP-Dokumentation nachgeschlagen werden. Mit Hilfe der Methode „ParseDIDLMessage“ wird die XML-Struktur in eine Objektstruktur überführt, in dem der XML-String sowie der entsprechende ParentContainer als Parameter an die Methode „ParseDIDLMessage“ übergeben werden. Durch Werfen des Events „OnResponseReceived“ wird der ContentManager frühzeitig über die Anzahl der zurückgelieferten Medienobjekte informiert.

Methode: BrowseMetadata Sichtbarkeit: public

Eingabe: CDSObject cdsObject, UInt32 startingIndex, UInt32 requestedCount

Ausgabe: void

Mit Hilfe dieser Methode wird eine Anforderung der Metadaten eines durch den Parameter „cdsobject“ angegebenen Medienobjektes durchgeführt. Unter Verwendung der entsprechenden „ContentDirectory“-Instanz wird so direkt mit der UPnP-AV-Implementierung kommuniziert. Diese Methode wurde bisher noch nicht verwendet und ist ungetestet.

Methode: BrowseMetadataSink Sichtbarkeit: private

Eingabe: ContentDirectory sender, String ObjectID, ContentDirectory.Enum_A_ARG_TYPE_BrowseFlag BrowseFlag, String Filter, UInt32 StartingIndex, UInt32 RequestedCount, String SortCriteria, String Result, UInt32 NumberReturned, UInt32 TotalMatches, UInt32 UpdateID, UPnPInvokeException ex, object _Tag

Ausgabe: void

Diese Methode nimmt die serverseitige Antwort zu einer „BrowseMetaData“-Anfrage entgegen und wird von der UPnP-AV-Implementierung nach Durchführung der Anfrage gerufen. Der String „ObjectID“ bezeichnet die ID des spezifischen Medienobjekts, dessen Metadaten als XML-konformer String in der Variablen „Result“ übergeben werden. Diese DIDL-Nachricht bedarf nun einer weiteren Aufbereitung, um die im DIDL-Format beschriebenen Medien in eine objektorientierte Struktur zu überführen. Die weiteren Parameter können in der UPnP-Dokumentation nachgeschlagen werden. Mit Hilfe der Methode „ParseDIDLMessage“ wird die XML-Struktur in eine Objektstruktur überführt.

Methode: ContentSession Sichtbarkeit: public

Eingabe: UPnPDevice device, String sortCriteria, String filter, SingleCDSObjectParsed callbackSingleCDSObjectParsed, AllCDSObjectsParsed callbackCompleteAllCDSObjectParsed, ErrorOccurred callbackErrorOccurred

Ausgabe: void

Bei dieser Methode handelt es sich um den Konstruktor für eine ContentSession-Instanz. Das übergebene „device“ entspricht dem Server und die Callbacks werden jeweils durch die Events „OnSingleCDSObjectParsed“, „OnAllCDSObjectParsed“ und „OnErrorOccurred“ aufgerufen. In diesem Fall entspricht dies den Methoden „SingleCDSObjectParsedSink“, „AllCDSObjectsParsedSink“ und „ErrorOccurredOnContentSessionSink“ im ContentManager. Diese Methoden wurden bereits im vorherigen Abschnitt besprochen.

Methode: ParseDIDLContainers Sichtbarkeit: private

Eingabe: XmlNodeList listContainers, CDSContainer cdsContainer

Ausgabe: void

Diese Methode besitzt eine ähnliche Implementierung wie die Methode „ParseDIDLItems“. Die Grundidee besteht darin, alle bekannten CDSContainer in eine Objektstruktur zu zerlegen. Übergeben wird eine XmlNodeList, die Container-Elemente enthält. Items werden mit Hilfe der Methode „ParseDIDLItems“ in eine objektorientierte Struktur überführt. Die Arbeitsweise dieser Methode ist recht interessant. Da ein DOM-Parser deutlich mehr Komfort bietet, wurde ein solcher verwendet, um die XMLstruktur in eine Baumstruktur zu überführen. Nachdem der Inhalt des Tags „upnp:class“ ausgelesen worden ist, muss nun das passende logische Objekt verwendet werden. Mit Hilfe einer Sammlung von Type-Instanzen wird per Reflection der Konstruktor der Klasse ermittelt, der ein Objekt bzgl. der im Tag „upnp:class“ angegebenen UPnP-Klasse erzeugt. Unter Verwendung der statischen Klassenvariable „upnpClassGeneral“ wird ein solches Mapping möglich gemacht. Wurde ein passender Konstruktor gefunden, so wird an diesen der XmlNode „container“ übergeben, um auf diese Weise ein Medien-Objekt erzeugen zu können. Eine nähere Beschreibung dieser Erzeugung ist in [1.1.2](#) zu finden. Vorab sei erwähnt, dass an dieser Stelle die in UPnP definierte CDSObject-Struktur per Vererbung nachgebildet wird, um mit Hilfe einer Konstruktorverkettung ein elegantes Parsen zu ermöglichen. Ist ein solches Medien-Objekt in Form einer CDSObject-Instanz erzeugt worden, so wird dieses per Event „OnSingleCDSObjectParsed“ an die GUI weitergeleitet.

Methode: ParseDIDLItems Sichtbarkeit: private

Eingabe: XmlNodeList listItems, CDSContainer cdsContainer

Ausgabe: void

Diese Methode besitzt eine ähnliche Implementierung wie die Methode „ParseDIDLContainers“. Wie dort werden alle bekannten CDSItems in eine Objektstruktur zu zerlegt. Übergeben wird eine XmlNodeList, die Item-Elemente enthält. Nachdem der Inhalt des Tags „upnp:class“ ausgelesen worden ist, wird mit Hilfe einer Sammlung von Type-Instanzen per Reflection der Konstruktor ermittelt der ein Objekt bzgl. der im Tag

„upnp:class“ angegebenen UPnP-Klasse erzeugt. Unter Verwendung der statischen Klassenvariable „upnpClassGeneral“ wird dieses Mapping ermöglicht. Wurde ein passender Konstruktor gefunden, wird an diesen der XmlNode „container“ übergeben, und auf diese Weise ein Medien-Objekt erzeugt. Eine nähere Beschreibung dieser Erzeugung ist in 1.1.2 zu finden. Vorab sei erwähnt, dass an dieser Stelle die in UPnP definierte CDSObject-Struktur per Vererbung nachgebildet wird, um mit Hilfe einer Konstruktorverkettung ein elegantes Parsen zu ermöglichen. Ist ein Medien-Objekt in Form einer CDSObject-Instanz erzeugt worden wird es per Event „OnSingleCDSObjectParsed“ an die GUI weitergeleitet.

Methode: ParseDIDLMessage Sichtbarkeit: private

Eingabe: String XML, CDSContainer cdsContainer

Ausgabe: void

Diese Methode leitet den Parse-Vorgang der empfangenen DIDL-Nachricht ein. Es wird eine Liste der unterstützten CDSObjekte erzeugt, die erweiterbar ist. Aus Performancegründen sollte dabei für spätere Zugriffe eine umgekehrte hierarchische Reihenfolge verwendet werden. Je weiter man also in den Listen „cdsTypesItems“ und „cdsTypesContainers“ fortschreitet, um so allgemeiner werden die Objektformate. Aus Komfortabilitätsgründen wird an dieser Stelle ein DOM-XML-Parser verwendet, der den String „XML“ in eine Baumstruktur zerlegt. Aus Effizienzgründen erfolgt eine Trennung zwischen Items und Containern. Da für das Finden des korrekten Konstruktors eine ineffiziente Iteration über die „Type“-Listen in Form von „cdsTypesItems“ bzw. „cdsTypesContainers“ notwendig ist, kann so die Gesamtanzahl der zu iterierenden Elemente nahezu halbiert werden. Im späteren Verlauf wäre es möglich, das Parsen in weitere Threads zu verlagern, um so einen weiteren Performance-Gewinn erreichen zu können. Mit Hilfe des Events „OnAllCDSObjectParsed“ wird signalisiert, dass die gesamte Nachricht erfolgreich geparsed wurde.

Methode: Search Sichtbarkeit: public

Eingabe: CDSContainer cdsContainer, String searchCriteria

Ausgabe: void

Mit Hilfe dieser Methode wird eine Suchanfrage an den jeweiligen Server gestellt. Unter Anwendung der passenden „ContentDirectory“-Instanz kann so direkt mit der UPnP-AV-Implementierung kommuniziert werden. Sollte eine Suchanfrage auf dem gewünschten Server nicht verfügbar sein, könnte eine spätere Implementierung dies durch ein rekursives „BrowseDirectChildren“ kompensieren. Diese Vorgehensweise ist jedoch enorm ineffizient und langsam, so dass auf eine Implementierung zunächst verzichtet wurde.

Methode: SearchSink Sichtbarkeit: private

Eingabe: ContentDirectory sender, String ContainerID, String SearchCriteria, String Filter, UInt32 StartingIndex, UInt32 RequestedCount, String SortCriteria, String Result, UInt32 NumberReturned, UInt32 TotalMatches, UInt32 UpdateID, UPnPInvokeException ex, object _Tag

Ausgabe: void

Die erhaltene Antwort der Suchanfrage ist in der Variablen „Result“ zu finden, die nun entsprechend geparsed werden muss, unter Verwendung der bereits vorgestellten Methode „ParseDIDLMessage“. Per Event „OnResponseReceived“ kann die Anzahl der erhaltenen Medienobjekte an den ContentManager, genauer, die Methode „OnNumberReturned“, weitergegeben werden. Die weiteren Parameter können in der UPnP-Spezifikation nachgeschlagen werden.

Klasse: DeviceManager Die Klasse „DeviceManager“ repräsentiert die dritte tragende Säule der „großen“ Manager-Elemente. Eine Instanz dieser Klasse übernimmt die gesamte Device-Verwaltung. Sie nimmt Ereignisse der Instanz „UPnPSmartControlPoint“ aus der UPnP-Implementierung von Intel in C# ? entgegen und leitet diese per Event „OnAddedDevice“, „OnAddedServer“, „OnAddedRenderer“, „OnRemovedDevice“, „OnRemovedServer“ oder „OnRemovedRenderer“ an interessierte Parteien weiter. Durch die Erweiterung der Architektur durch den Follow-Me-Modus erhielt diese Klasse weiteres Gewicht. Mit Hilfe des Events „OnNearestCompatibleRenderer“ erfolgt die Information über den Wechsel des Renderers der zum User am nächsten gelegenen ist. Bisher ist es nur möglich eine einzige AVSession im Follow-Me-Modus zu betreiben. Ein umfassendes Konflikt-Management wäre ansonsten notwendig, um Mehrdeutigkeiten im Konfliktfall lösen zu können.

Variable: nearestRenderers Sichtbarkeit: private

Diese Variable speichert in einer ArrayList-Instanz 2er Tupel (DeviceRSSIValue-Instanzen) bestehend aus dem RSSI-Wert und der UPnPDevice-Instanz. Innerhalb dieser Liste sind alle diejenigen Wiedergabe-Geräte gelistet, die zur Zeit dem Nutzer am nächsten sind und einen RSSI-Wert von größer gleich 0 besitzen.

Variable: readRSSIValue Sichtbarkeit: private

Typ: Thread

Diese Variable wird verwendet um innerhalb des DeviceManagers weiterhin Zugriff auf einen eventuell offenen Thread zu haben, der zum aktuellen Zeitpunkt die RSSI-Werte der bestehenden Bluetooth-Verbindungen ausliest. Die Referenz auf diesen Thread, in dem die Methode „GetNearestRenderer“ ausgeführt wird, hält diese Variable.

Variable: selectedRenderer Sichtbarkeit: private

Typ: UPnPDevice

Innerhalb dieser Variablen wird der zur Zeit in der GUI ausgewählte Renderer in Form einer UPnPDevice-Instanz gespeichert.

Variable: startBluetoothSearch Sichtbarkeit: private

Typ: Thread

Diese Variable wird verwendet um innerhalb des DeviceManagers weiterhin Zugriff auf

einen eventuell offenen Thread zu haben, der zum aktuellen Zeitpunkt den Bluetooth-Scan durchführt. Die Referenz auf diesen Thread, in dem die Methode „StartBluetoothSearch“ ausgeführt wird, hält diese Variable.

Event: OnAddedDevice Eingabe: UPnPDevice device

Ausgabe: void

Dieses Event informiert darüber, dass ein neues Device (ein Renderer oder Server) verfügbar ist und zur Nutzung bereit steht.

Event: OnAddedOtherUPnPDevice Eingabe: UPnPDevice device

Ausgabe: void

Dieses Event informiert darüber, dass ein neues Device verfügbar ist. Dieses Event wird zur Zeit nicht verwendet und verlangt eine Modifikation des Suchkriteriums in der vorliegenden UPnP-Implementierung.

Event: OnAddedRenderer Eingabe: UPnPDevice device

Ausgabe: void

Dieses Event informiert darüber, dass ein neuer Renderer verfügbar ist und zur Nutzung bereit steht.

Event: OnAddedServer Eingabe: UPnPDevice device

Ausgabe: void

Dieses Event informiert darüber, dass ein neuer Server verfügbar ist und zur Nutzung bereit steht.

Event: OnDebugInfo Eingabe: String text

Ausgabe: void

Dieses Event wird für das Debugging verwendet, indem Debug-Informationen an die GUI weitergeleitet werden.

Event: OnNearestCompatibleRenderer Eingabe: UPnPDevice device, CDSObject cdsObject, String useProtocol, Uri avTransportURI

Ausgabe: void

Dieses Event informiert den AVSession-Manager darüber, dass eine neue AVSession zu dem angegebene Device aufzubauen ist, unter Verwendung des Protokolls „useProtocol“ und der „avTransportURI“. Ist das entsprechende Medienobjekt bekannt, so wird dieses ebenfalls weitergegeben. Dieses Event wird nur dann geworfen, wenn sich der Renderer, in dessen unmittelbarer Nähe sich der Nutzer befindet, auch tatsächlich verändert hat und dieser kompatibel zu dem Medien-Objekt ist, das sich momentan im Follow-Me-Modus befindet. Gerufen wird die Methode „NearestCompatibleRenderer“ im AVSessionManager.

Event: OnRemovedDevice Eingabe: UPnPDevice device

Ausgabe: void

Dieses Event informiert darüber, dass ein Device (ein Renderer oder Server) nicht länger verfügbar ist.

Event: OnRemovedOtherUPnPDevice Eingabe: UPnPDevice device

Ausgabe: void

Dieses Event informiert darüber, dass ein Device nicht länger verfügbar ist. Dieses Event wird zur Zeit nicht verwendet und verlangt eine Modifikation des Suchkriteriums in der vorliegenden UPnP-Implementierung.

Event: OnRemovedRenderer Eingabe: UPnPDevice device

Ausgabe: void

Dieses Event informiert darüber, dass ein Renderer nicht mehr verfügbar ist.

Event: OnRemovedServer Eingabe: UPnPDevice device

Ausgabe: void

Dieses Event informiert darüber, dass ein Server nicht mehr verfügbar ist.

Methode: DeviceAdded Sichtbarkeit: public

Eingabe: UPnPSmartControlPoint sender, UPnPDevice device

Ausgabe: void

Diese Methode wird per Event „OnAddedDevice“ aus der UPnPSmartControlPoint-Instanz heraus ausgerufen. Diese Methode wirft nun individuelle Events per „OnAddedServer“ oder „OnAddedRenderer“. Es sei erwähnt, dass diese drei Events nicht ganz gradlinig verwendet werden. Zukünftige Versionen könnte eine klarere Struktur in diesem Bereich aufweisen.

Methode: DeviceManager Sichtbarkeit: public

Eingabe: DeviceHandler callbackDeviceHandler1, DeviceHandler callbackDeviceHandler2

Ausgabe: void

Hierbei handelt es sich um den Konstruktor der Klasse „DeviceManager“, mit dessen Hilfe Instanzen des DeviceManagers erzeugt werden. Die beiden übergebenen Methodenreferenzen, registrieren die Methode „DeviceAdded“ im ContentManager sowie die Methode „DeviceAdded“ im AVSessionManager.

Methode: Dispose Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Mit Hilfe dieser Methode können „Aufräum“-Operationen durchgeführt werden. Durch den Aufruf von „StopGetNearestRenderer“ wird der bestehende Follow-Me-Modus unterbrochen. Der Abbruch aller geworfenen Events ist eine nicht notwendige Bedingung,

die aber der Vollständigkeit halber durchgeführt werden sollte, um Abbruch-Fehler zu vermeiden.

Methode: GetAllRenderers Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Unter Verwendung dieser Methode werden alle zur Zeit bekannten Renderer-Objekte per Event „OnAddedRenderer“ an interessierende Parteien geworfen. Zurückgegriffen wird an dieser Stelle auf die interne Liste der UPnP SmartControlPoint-Instanz.

Methode: GetAllRenderers Sichtbarkeit: public

Eingabe: DeviceHandler callbackRendererAdded

Ausgabe: void

Unter Verwendung dieser Methode werden alle zur Zeit bekannten Renderer-Objekte durch einen Methoden-Aufruf der im Parameter angegebenen Methode ausgegeben. Zurückgegriffen wird an dieser Stelle auf die interne Liste der UPnP SmartControlPoint-Instanz.

Methode: GetAllRenderersList Sichtbarkeit: public

Eingabe: void

Ausgabe: ArrayList

Mit Hilfe dieser Methode kann eine Array-Liste aller bekannten Renderer-Objekte im UPnP-Netzwerk angefordert werden.

Methode: GetAllServers Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Unter Verwendung dieser Methode werden alle zur Zeit bekannten Server-Objekte per Event „OnAddedRenderer“ gemeldet. Zurückgegriffen wird an dieser Stelle auf die interne Liste der UPnP SmartControlPoint-Instanz.

Methode: GetAllServers Sichtbarkeit: public

Eingabe: DeviceHandler callbackRendererAdded

Ausgabe: void

Unter Verwendung dieser Methode werden alle zur Zeit bekannten Server-Objekte durch Aufruf der im Parameter angegebenen Methode ausgegeben. Zurückgegriffen wird an dieser Stelle auf die interne Liste der UPnP SmartControlPoint-Instanz.

Methode: GetAllServersList Sichtbarkeit: public

Eingabe: void

Ausgabe: ArrayList

Mit Hilfe dieser Methode kann eine Array-Liste aller bekannten Server-Objekte im UPnP-Netzwerk angefordert werden.

Methode: GetDevicePerUUID Sichtbarkeit: public

Zugriffsmodifier: static

Eingabe: string uuid

Ausgabe: UPnPDevice

Diese statische Methode erlaubt es, unter Verwendung der im Parameter „uuid“ angegebenen UUID das passende logische UPnPDevice-Objekt zu erhalten. Diese Methode wurde nachträglich hinzugefügt und wird dafür verwendet, um aus der in UPnP definierten „PeerConnectionManager“-Variablen auf die zugehörige zweite Partei schließen zu können.

Methode: GetNearestRenderer Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Diese Methode benötigt eine detaillierte Erklärung. Sie wird nur für den Follow-Modus verwendet. Es erwies sich als eine recht schwierige Aufgabe, den RSSI für eine Verbindung auszulesen, daher musste an dieser Stelle „getrickst“ werden. Da es sich dabei aber nur zu Testzwecken entwickeltes Szenario handelt, muss eine spätere Implementierung, ein Redesign des Follow-Me-Modus vornehmen.

Verwendet werden zwei ArrayList-Instanzen „tmpNearestRenderer“ sowie „NearestRenderers“. Die Liste „NearestRenderers“, soll später „DeviceRSSIValue“-Instanzen aufnehmen, die ein 2er Tupel bestehend aus RSSI-Wert und dem Renderer in Form einer UPnPDevice-Instanz darstellen. Die Liste „tmpNearestRenderer“ soll „UUIDRSSIValue“-Instanzen enthalten, die ebenfalls 2er Tupel darstellen, bestehend aus dem RSSI-Wert sowie der Geräte-Kennung in Form einer UUID. Um den Inhalt dieser Liste erzeugen zu können, wird in einer nicht-terminierenden Schleife alle 5 Sekunden eine Text-Datei „rssidata.txt“ ausgelesen. Diese Textdatei enthält per Zeilenumbruch getrennte Einträge. Ein Eintrag entspricht der Form „UUID:RSSI“. Die „UUID“ entspricht der UUID im Sinne der UPnP-Spezifikation und ermöglicht auf diese Weise, die konkrete UPnPDevice-Instanz zu ermitteln. Durch „:“ getrennt kann nun der RSSI-Wert der Bluetooth-Verbindung ausgelesen werden. Es handelt sich hierbei um einen „Double“-Wert z.B. 3.66667, der jedoch auch negativ sein kann. Konnte der Wert nicht ausgelesen werden, so beträgt der RSSI-Wert für dieses Device -1000 . Da zunächst ein länger andauernder Scan nach Devices notwendig ist, wird per „istFertig()“ das Scan-Ende signalisiert. Da ein RSSI-Wert 0 eine „optimale“ Bluetooth-Verbindung beschreibt, kann es durchaus vorkommen, dass mehrere Geräte diesen Wert aufweisen. In diesem besonderen Ausnahmefall werden auf beiden Geräte neue Medien-Wiedergaben initiiert. Nach dem vollständigen Auslesen der Text-Datei enthält die Liste „tmpNearestRenderer“ alle diejenigen Tupel, deren RSSI-Wert maximal ist. Wie bereit angesprochen, kann die Liste mehr als einen Eintrag enthalten. Ist der maximale RSSI-Wert mindestens ≥ 0 , so folgt eine Iteration durch die Liste „tmpNearestRenderer“, wobei für jeden Eintrag – durch den Aufruf der Methode „GetRendererPerUUID“ – die UPnPDevice-Instanz ermittelt wird.

Methode: GetRendererPerUUID Sichtbarkeit: public

Eingabe: string uuid, double rssiValue

Ausgabe: void

Mit Hilfe dieser Methode wird das zu der im Parameter übergebenen UUID gehörige Device-Objekt ermittelt, indem die interne Liste der UPnP SmartControlPoint-Instanz durchlaufen wird. Konnte nun die UPnP Device-Instanz erfolgreich ermittelt werden, so wird das Event „OnNearestCompatibleRenderer“ geworfen. Aufgerufen wird auf diese Weise die Methode „NearestCompatibleRenderer“ im AVSession-Manager, die automatisch eine neue Verbindung zu dem übergebenen Renderer herstellen soll. An dieser Stelle stoßen wir jedoch an sehr enge Grenzen. Das bereits angesprochene Dilemma, das es nicht immer möglich, die Quelle eine Verbindung zu identifizieren, erweist sich als großes Problem. Es ist maximal möglich, das für den Transfer eingesetzte Protokoll vom Renderer per UPnP-Aktion „GetCurrentConnectionInfo“ (siehe ?) zu erfahren. Diese Information ist bereits intern verfügbar, da sie bereits bei der Erzeugung einer AVSession-Instanz in der Variablen „usedProtocol“ gespeichert wurde. Es kann unterschieden werden, ob es sich um eine selbst-initiierte Verbindung handelt oder nicht. Im ersten Fall können Alternativ-Protokolle verwendet werden, da Quelle, Senke und Medien-Objekt vollständig bekannt sind. In dieser Referenz-Implementierung wird dieses Feature zur Zeit allerdings nicht unterstützt. Es erfolgt die Verwendung des in „usedProtocol“ eingetragenen Transferprotokolls. Im anderen Fall steht nur die Information vom Renderer zur Verfügung, die zu Problemen führen kann. Im Falle des Intel-Renderers ? wird die Information über das verwendete Protokoll nur dann gespeichert, wenn die Methode „PrepareForConnection“ vorhanden ist, ansonsten erhält man nur den String „:“ mit dessen Hilfe keine Protokoll-Ermittlung möglich ist. Daher sollte Folgendes beachtet werden: Ein Follow-Modus sollte am Besten auf Sessions beschränkt sein, die vom Nutzer selbst-initiiert wurden, da der gesamte Informationsgehalt zur Verfügung steht mit dessen Hilfe eine adäquate Entscheidung über das zu verwendende Transferprotokoll getroffen werden kann. Zu diesem Zeitpunkt ist jedoch auch eine bereits aktive Verbindung in den Follow-Me-Modus versetzbar, falls die Protokoll-Information dieser Verbindung zur Verfügung gestellt wurde. Dieses Feature wurde jedoch noch nicht getestet. Es muss darauf hingewiesen werden, dass der „Follow-Me“-Modus in dieser Form nicht als ausgereift oder sicher betrachtet werden kann. In Testszenarien konnte die Arbeitsweise dieses Modus bisher nicht überzeugen.

Methode: RemovedDevice Sichtbarkeit: public

Eingabe: UPnP SmartControlPoint sender, UPnP Device device

Ausgabe: void

Diese Methode wird per Event „OnRemovedDevice“ aus der UPnP SmartControlPoint-Instanz heraus ausgerufen. Diese Methode wirft nun individuelle Events per „OnRemovedServer“ oder „OnRemovedRenderer“.

Methode: StartBluetoothSearch Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Diese Methode startet einen Scan nach allen verfügbaren Bluetooth-Geräten. Da sich diese gefundene Gerätekonstellation dynamisch verändern kann, indem neue Geräte hinzukommen oder alte nicht mehr verfügbar sind, erfolgt ein Scan nach verfügbaren Bluetooth-Geräten alle 30 Sekunden, falls der Follow-Me-Modus aktiviert ist. Verwendet wird hierfür die Methode „rascan“. Um überlappende Suchvorgänge zu vermeiden, wird ein neuer Suchvorgang nur dann initiiert, wenn ein vorheriger Suchvorgang erfolgreich beendet werden konnte.

Methode: StartGetNearestRenderer Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Diese Methode startet den Bluetooth-Scan nach dem Gerät mit der kürzesten Entfernung zum Nutzer durch Aufruf der Methode „StartBluetoothSearch“ in einem separaten Thread. Ebenso wird ein weiterer Thread verwendet, in dem die Methode „GetNearestRenderer“ ausgeführt wird. Grundlage dafür ist der RSSI-Wert einer bestehenden Verbindung. Da das Auslesen aus einer Text-Datei recht lange dauern kann, muss die dafür verantwortliche Methode „GetNearestRenderer“ in einem separaten Thread gestartet werden. Diese Methode wird dafür verwendet, um denjenigen bzw. diejenigen Renderer zu ermitteln, der bzw. die dem Nutzer zur Zeit am nächsten ist bzw. sind.

Methode: StopGetNearestRenderer Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Da sich stets nur eine Verbindung im Follow-Me-Modus befinden kann, wird das aufwendige Auslesen der Text-Datei nur dann durchgeführt, wenn der Follow-Me-Modus auch tatsächlich aktiviert ist. In diesem Fall werden die beiden Threads „startBluetoothSearch“ und „readRSSIValue“ per „Abort()“ abgebrochen. Ebenso wird per „deviceclose()“ alle vorhandenen Bluetooth-Verbindungen sauber beendet.

Methode: StopThreadBluetoothSearch Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Durch den Aufruf dieser Methode wird derjenige Thread beendet, der eine Bluetooth-Suche nach neuen Geräten durchführt.

private: StopThreadRSSIValue Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Durch den Aufruf dieser Methode wird derjenige Thread beendet, der den oder die Renderer ermitteln soll, die dem Nutzer zur Zeit am nächsten sind.

Klasse: MainForm Diese Klasse stellt die GUI zur Verfügung über die eine Interaktion mit dem Nutzer erfolgt. Die Funktionalität, die diese Klasse zur Verfügung steht ist enorm komplex, da umfangreiche nebenläufige GUI-Modifikationen notwendig sind, um die komplexen Abläufe innerhalb des UPnP-basierten Netzwerks abbilden zu können. Die GUI besteht aus fünf Hauptkategorien: „Browse“, „Suche“, „Alle Audio-Files“, „Alle Video-Files“, „Vorhandene AVSessions“ und „Debugging“.

Methode: ActiveAVSessionSwitched Sichtbarkeit: private

Eingabe: AVSession avSession

Ausgabe: void

Diese Methode wird durch das Event „OnActiveAVSessionSwitched“ im AVSession-Manager aufgerufen und informiert die GUI darüber, dass nun die im Parameter „avSession“ angegebene „AVSession“-Instanz die aktive Verbindung beschreibt. Die Ausführung wird an den Thread delegiert, der das Control „treeViewRenderer“ erzeugt hat. Dabei wartet „BeginInvoke“ nicht bis der delegierte Code ausgeführt ist.

Methode: ActiveAVSessionSwitchedSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs args

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein Update des Players mit dessen Hilfe eine aktive Steuerung der neuen Session möglich ist.

Methode: ActiveRendererInfo Sichtbarkeit: private

Eingabe: UPnPDevice device, CommonStateInfo stateInfo

Ausgabe: void

Diese Methode wird durch das Event „OnActiveRendererInfo“ im AVSession-Manager aufgerufen und leitet Informationen in Form von „CommonStateInfo“-Instanzen an die GUI weiter. Diese Informationen werden nur dann angezeigt, wenn innerhalb der GUI ein Renderer und keine AVSession ausgewählt wurde. Die Ausführung wird an den Thread delegiert, der das Control „listViewInfo“ erzeugt hat.

Methode: ActiveRendererInfoSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs args

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt eine Aktualisierung der Renderer-bezogenen Informationen, die z. B. die unterstützten Transferprotokolle sowie die IDs der zur Zeit aktiven Verbindungen umfassen.

Methode: AVSessionComplete Sichtbarkeit: private

Eingabe: UPnPDevice device, AVSession avSession

Ausgabe: void

Diese Methode wird durch das Event „OnAVSessionComplete“ im AVSession-Manager aufgerufen und leitet die Information der beendeten Medien-Wiedergabe in Form einer

„AVSession“-Instanz an die GUI weiter. Die Ausführung wird an den Thread delegiert, der das Control „treeViewRenderer“ erzeugt hat.

Methode: AVSessionCompleteSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs args

Ausgabe: void

Mit Hilfe dieser Methode wird die Baum-Ansicht, die die Renderer und die offenen Medien-Wiedergaben umfasst aktualisiert, indem die betroffene „AVSession“-Instanz in Form eines „TreeNodeAVSession“-Objekts aus der Ansicht entfernt wird.

Methode: AVSessionInformation Sichtbarkeit: private

Eingabe: UPnPDevice device, AVSession avSession, CommonStateInfo stateInfo

Ausgabe: void

Diese Methode wird durch das Event „OnAVSessionInfo“ im AVSession-Manager aufgerufen und leitet Informationen in Form von „CommonStateInfo“-Instanzen an die GUI weiter. Zum aktuellen Implementierungszeitpunkt findet keine Filterung der Informationen statt. Verwendet wird zur Zeit allein die Information, die den durch „TransportState“ bezeichneten internen Zustand des Renderers beschreibt.

Methode: AVSessionInformationSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs args

Ausgabe: void

Mit Hilfe dieser Methode wird die Baum-Ansicht, die die Renderer und die offenen Medien-Wiedergaben umfasst aktualisiert, indem die betroffene AVSession-Instanz, welche mit Hilfe des Parameters „obj“ getunnelt wurde, verwendet wird, um die entsprechende Information zu aktualisieren. Sollte eine AVSession noch nicht bekannt sein, so wird ein neuer Baumknoten erzeugt und in den Baum eingehängt.

Methode: btnAudio_Click Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird durch einen Klick auf den Button „Alle Audio-Files“ aufgerufen. Erzeugt wird eine neues ListView-Objekt sowie ein Aufruf der Methode „GetAllAudioFiles“ im Content-Manager, der eine netzweite Suche nach allen verfügbaren Audio-Files einleitet.

Methode: btnBrowse_Click Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird durch einen Klick auf den Button „Browse“ aufgerufen. Erzeugt wird eine neues ListView-Objekt sowie ein Aufruf der Methode „ShowBrowseServers“, mit dessen Hilfe alle verfügbaren Server in der GUI angezeigt werden.

Methode: btnDebug_Click Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird durch einen Klick auf den Button „Debugging“ aufgerufen und ermöglicht auf diese Weise den Zugang zur Debugging-Konsole, die zur Zeit nur Informationen, die den „Follow-Me“-Modus betreffen, anzeigt.

Methode: btnPause_Click Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird durch einen Klick auf den Button „Pause“ aufgerufen und versetzt den Zustand der aktuell steuerbaren Medien-Verbindung durch den Aufruf der Methode „Pause“ im AVSession-Manager in den Status „PAUSED_PLAYBACK“.

Methode: btnPicture_Click Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird durch einen Klick auf den Button „Alle Picture-Files“ aufgerufen. Erzeugt wird eine neues ListView-Objekt sowie ein Aufruf der Methode „GetAllPictures“ im Content-Manager, der eine netzweite Suche nach allen verfügbaren Bildern einleitet.

Methode: btnPlay_Click Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird durch einen Klick auf den Button „Play“ aufgerufen und versetzt den Zustand der aktuell steuerbaren Medien-Verbindung durch den Aufruf der Methode „Play“ im AVSession-Manager in den Status „PLAYING“.

Methode: btnRenderer_Click Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Durch den Aufruf dieser Methode werden die in der Baum-Ansicht vorhandenen Renderer- und AVSession-Informationen verworfen und durch den Aufruf der Methode „GetCurrentAVSessionsInfo“ im AVSession-Manager neu angefordert. Diese Informationen müssten zum aktuellen Implementierungsstand nicht verworfen werden, da sie stets aktualisiert werden, auch wenn die aktuelle Ansicht nicht der Renderer-Darstellung entspricht. Dennoch bestand die Veranlassung, im Falle eines Verlassens dieser Renderer-Darstellung, das Update dieser GUI-Ansicht vollständig einzustellen, um so ressourcensparender Arbeiten zu können. Ein solches Event-„Canceling“ wird zur Zeit jedoch nicht aktiv unterstützt.

Methode: btnSearch_Click Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird durch einen Klick auf den Button „Suche“ aufgerufen. Erzeugt wird die bisher in ihren Fähigkeiten stark eingeschränkte Such-Maske, die durch den Aufruf der Methode „ConfigureSearchForm“ auf die aktuell vorherrschende Server-Situation dynamisch angepasst wird.

Methode: btnStop_Click Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird durch einen Klick auf den Button „Stop“ aufgerufen und versetzt den Zustand der aktuell steuerbaren Medien-Verbindung durch den Aufruf der Methode „Stop“ im AVSession-Manager in den Status „STOPPED“.

Methode: btnVideo_Click Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird durch einen Klick auf den Button „Alle Video-Files“ aufgerufen. Erzeugt wird eine neues ListView-Objekt sowie ein Aufruf der Methode „GetAllVideo-Files“ im Content-Manager, der eine netzweite Suche nach allen verfügbaren Video-Files einleitet.

Methode: checkBoxFilter_CheckStateChanged Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird aufgerufen, sobald sich der Zustand der Filter-CheckBox ändert. Durch den Aufruf der Methode „FilterCurrentContent“ wird die aktuelle Medien-Ansicht aktualisiert, indem alle angezeigten Medien-Objekte auf Kompatibilität zu dem ausgewählten Renderer geprüft werden. Diese Methode verarbeitet auch eine Filter-Deaktivierung, indem alle verfügbaren Medien-Objekte an die GUI gesendet werden.

Methode: checkBoxFollowMe_CheckStateChanged Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird aufgerufen, sobald sich der Zustand der Follow-Me-CheckBox ändert. Mit Hilfe diverser Überprüfungen wird festgestellt, ob sich die Veränderung des CheckBox-Zustandes auf die aktuelle AVSession bezieht oder nicht, da nur in diesem Fall eine Deaktivierung bzw. Aktivierung notwendig ist.

Methode: ConfigureSearchForm Sichtbarkeit: private

Eingabe: int matchesTo

Ausgabe: void

Durch den Aufruf dieser Methode werden alle Suchfunktionen, die von dem oder den

gewählten Servern unterstützt werden, aktiviert oder deaktiviert. Soll eine netzweite Suche durchgeführt werden, so erfolgt eine Schnittmengen-Bildung über alle verfügbaren Such-Kriterien. Die durch „matchesTo“ notwendige Anzahl positiver Matches eines Suchkriteriums, das zu einer Aktivierung führt, wird als Parameter an diese Methode übergeben.

Methode: contentListView_ItemActivate Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird aufgerufen, wenn in ein Medien-Objekt, -Container oder Server per Doppel-Klick ausgewählt wurde. Es erfolgt eine serverseitige „Browse“-Anfrage oder aber die Initiierung eines Abspielvorgangs auf dem bereits selektierten Server. Falls noch keine Server ausgewählt wurde, so wird mit Hilfe eines Dialogs der Nutzer aufgefordert, einen Renderer auszuwählen.

Methode: contentQuickNavigationBox_SelectedIndexChanged Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird aufgerufen, wenn innerhalb der ComboBox, die eine Navigation ermöglicht, eine Veränderung auftritt. Auf diese Weise wird ein Datei-Explorer realisiert. Es ist anzumerken, dass bereits besuchte Verzeichnis-Hierarchien durch eine Serveranfrage neu angefordert werden. Komplexe Caching-Verfahren könnten an dieser Stelle verwendet werden, um die Reaktionszeiten beim Browse-Vorgang zu beschleunigen.

Methode: contextMenuContent_Popup Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird aufgerufen, bevor das Kontext-Menü für ein Medien-Objekt, -Container oder Server aufgerufen wird. Auf diese Weise kann ein Item-individuelles Kontext-Menü erzeugt werden, das in der Lage ist, sich der aktuellen Netz-Situation anzupassen, um so dem Nutzer einen maximalen Komfort zu bieten.

Methode: contextMenuItemContent_Click Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird aufgerufen, wenn der Nutzer einen Eintrag innerhalb des durch den Aufruf der Methode „contextMenuContent_Popup“ erzeugten Kontext-Menüs ausgewählt hat. Je nach Auswahl wird entweder ein Abspiel-Vorgang auf dem gewünschten Renderer initiiert oder aber die „Eigenschaften“ des gewählten Mediums angezeigt.

Methode: contextMenuItemRenderer_Click Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird aufgerufen, wenn der Nutzer einen Eintrag innerhalb des durch

den Aufruf der Methode „contextMenuRenderer_Popup“ erzeugten Kontext-Menüs ausgewählt hat. Je nach Auswahl wird entweder die selektierte AVSession-Instanz beendet bzw. gestoppt oder aber eine weitere Netzerkundung durchgeführt.

Methode: contextMenuRenderer_Popup Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird aufgerufen, bevor das Kontext-Menü für einen Renderer oder eine Medien-Wiedergabe aufgerufen wird. Ein AVSession-Abbruch wird nur erlaubt, wenn der Renderer die Methode „PrepareForConnection“ anbietet, ansonsten kann die Wiedergabe nur gestoppt werden.

Methode: DebugTextInfo Sichtbarkeit: private

Eingabe: String text

Ausgabe: void

Diese Methode wird durch das Event „OnDebugInfo“ im Content-Manager aufgerufen und leitet die Information der beendeten Medien-Wiedergabe in Form eines Strings an die GUI weiter. Die Ausführung wird an den Thread delegiert, der das Control „textBoxDebug“ erzeugt hat.

Methode: DebugTextInfoSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs args

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt eine Ausgabe in einer Debugging-Konsole des im Objekt „obj“ getunnelten Strings.

Methode: ErrorOccurredOnAVSession Sichtbarkeit: private

Eingabe: AVSession avSession, UPnPInvokeException ex

Ausgabe: void

Trat innerhalb einer AVSession ein Fehler auf, so wird diese Methode aufgerufen, indem die betroffene Medien-Wiedergabe in Form einer „AVSession“-Instanz und der aufgetretene Fehler übergeben werden. Die Ausführung wird an den Thread delegiert, der das Control „statusBarContent“ erzeugt hat.

Methode: ErrorOccurredOnAVSessionSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs e

Ausgabe: void

Innerhalb dieser Methode wird der Player zurückgesetzt, falls sich der Fehler auf die aktuelle AVSession bezog und eine Ausgabe in der Status-Bar erzeugt.

Methode: ErrorOccurredOnContentSession Sichtbarkeit: private

Eingabe: ContentSession contentSession, UPnPInvokeException ex

Ausgabe: void

Trat innerhalb einer ContentSession ein Fehler auf, so wird diese Methode aufgerufen, indem die betroffene Verbindung in Form einer „ContentSession“-Instanz und der aufgetretene Fehler übergeben werden. Die Ausführung wird an den Thread delegiert, der das Control „statusBarContent“ erzeugt hat.

Methode: ErrorOccurredOnContentSessionSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs e

Ausgabe: void

Innerhalb dieser Methode wird eine Ausgabe in der Status-Bar erzeugt.

Methode: GenerateContextMenuContent Sichtbarkeit: private

Eingabe: UPnPDevice device, CDSObject cdsObject

Ausgabe: void

Durch den Aufruf dieser Methode wird das anzuzeigende Kontext-Menü für ein Medien-Objekt generiert und konfiguriert. Es ermöglicht die Auswahl eines kompatiblen Renderers für eine Medien-Wiedergabe.

Methode: GenerateGuild Sichtbarkeit: private

Eingabe: void

Ausgabe: int

Mit Hilfe dieser Methode wird eine Zufallszahl erzeugt, die genutzt wird, um eine GUI-„Ansicht“ mit einem eindeutigen Wert zu versehen. Diese Identifikation wird benötigt, um angemeldete Events der passenden „Ansicht“ zuordnen zu können.

Methode: GenerateListView Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Diese Methode erzeugt ein ListView-Objekt, das für die Darstellung der Medien-Objekte, -Container und Server verwendet wird.

Methode: GenerateListViewInfo Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Diese Methode erzeugt ein ListView-Objekt, das für die Darstellung der CommonStateInfo-Objekte verwendet wird. Diese ListView-Instanz stellt die zu einer laufenden AVSession bekannten Informationen dar wie z.B. die relative Abspielzeit oder das Lautstärke-Niveau.

Methode: getContentIcon Sichtbarkeit: private

Eingabe: string UPnPClass

Ausgabe: int

Diese Methode liefert der Eingabe einer UPnP-Klasse den Index des passenden Icons zurück.

Methode: MainForm Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor der Klasse „MainForm“. Durch den Aufruf dieses Konstruktors wird die gesamte GUI erzeugt und konfiguriert.

Methode: MainForm_Closed Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird aufgerufen, bevor das Hauptfenster geschlossen wird. Es werden eine Vielzahl von Aufräum-Operationen durchgeführt, um z.B. noch aktive Threads zu beenden.

Methode: MainForm_KeyDown Sichtbarkeit: private

Eingabe: object sender, KeyEventArgs e

Ausgabe: void

Wird zur Zeit nicht verwendet.

Methode: NumberReturned Sichtbarkeit: private

Eingabe: ContentSession contentSession, Int32 numberReturned

Ausgabe: void

Diese Methode wird durch das Event „OnResponseReceived“ im Content-Manager aufgerufen und beinhaltet die Anzahl der gefundenen Treffer. Die Ausführung wird an den Thread delegiert, der das Control „statusBarContent“ erzeugt hat.

Methode: NumberReturnedSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs args

Ausgabe: void

Diese Methode erzeugt eine Ausgabe in der Status-Leiste innerhalb der GUI.

Methode: RendererAdded Sichtbarkeit: private

Eingabe: UPnPDevice device

Ausgabe: void

Aufgerufen wird diese Methode durch eine Parameter-Übergabe an die Methode „GetAllRenderers“ in Form eines Methodenzeigers auf diese Methode. Die Ausführung wird an den Thread delegiert, der das Control „contentListView“ erzeugt hat.

Methode: RendererAddedSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs args

Ausgabe: void

Diese Methode erzeugt das Kontext-Menü, das dem Nutzer die Möglichkeit bietet, eine Medien-Wiedergabe einzuleiten oder Meta-Informationen zu einem Medien-Objekt anzufordern.

Methode: ServerAdded Sichtbarkeit: private

Eingabe: UPnPDevice device

Ausgabe: void

Diese Methode wird durch das Event „OnAddedServer“ im Device-Manager aufgerufen und übergibt an die GUI einen neuen Server in Form einer UPnPDevice-Instanz. Die Ausführung wird an den Thread delegiert, der das Control „contentListView“ erzeugt hat.

Methode: ServerAddedSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs e

Ausgabe: void

Diese Methode aktualisiert – falls notwendig – die GUI-Ansicht und fügt den durch „obj“ angegebenen Server hinzu.

Methode: ServerRemoved Sichtbarkeit: private

Eingabe: UPnPDevice device

Ausgabe: void

Diese Methode wird durch das Event „OnRemovedServer“ im Device-Manager aufgerufen und übergibt an die GUI einen Server in Form einer UPnPDevice-Instanz, der nicht mehr verfügbar ist. Die Ausführung wird an den Thread delegiert, der das Control „contentListView“ erzeugt hat.

Methode: ServerRemovedSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs e

Ausgabe: void

Diese Methode aktualisiert – falls notwendig – die GUI-Ansicht und entfernt den durch „obj“ angegebenen Server.

Methode: serverSearchField_SelectedIndexChanged Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird aufgerufen, wenn in der Such-Maske ein Server oder „Alle Server“ ausgewählt wurde. Es erfolgt eine Anpassung der Such-Maske durch den Aufruf der Methode „ConfigureSearchForm“.

Methode: SetFocusTo Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Durch den Aufruf dieser Methode wird die zur Zeit steuerbare AVSession-Instanz innerhalb der Baum-Ansicht fokussiert.

Methode: ShowBrowseServers Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Durch den Aufruf dieser Methode werden alle zur Zeit verfügbaren Server durch den Aufruf der Methode „GetAllServers“ im Content-Manager angefordert.

Methode: ShowMessageBox Sichtbarkeit: private

Zugriffsmodifier: static

Eingabe: string text, string title

Ausgabe: void

Durch den Aufruf dieser Methode kann ein Dialog-Fenster mit dem durch „title“ angegebenen Titel und den durch „text“ angegebenen Text angezeigt werden.

Methode: startSearchButton_Click Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird aufgerufen, wenn der Benutzer einen Such-Vorgang initiiert. Durch den Aufruf der Methode „GetSearchResults“ im Content-Manager wird eine Anfrage an den oder die gewünschten Server gestellt.

Methode: trackBarVolume_ValueChanged Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Ist der Schieberegler, der die Lautstärke der aktuellen AVSession repräsentiert, vom Nutzer fokussiert, so wird das modifizierte Lautstärke-Level an den Renderer geschickt. Es ist zu unterscheiden, ob die Anpassung aktiv vom Nutzer erzeugt wird oder aber durch einen anderen Kontrollpunkt verursacht wird, um überschneidende und widersprüchliche Lautstärke-Levels ausschließen zu können.

Methode: treeViewRenderer_AfterSelect Sichtbarkeit: private

Eingabe: object sender, TreeViewEventArgs e

Ausgabe: void

Wurde innerhalb der Baum-Ansicht, die Informationen über die vorhandenen Renderer sowie zur Zeit aktive Medien-Wiedergaben enthält, ein Knoten ausgewählt, so wird die „ListView“-Instanz „listViewInfo“ dafür verwendet, um detaillierte Informationen zu dem selektierten Baum-Knoten anzuzeigen. Durch den Aufruf der Methode „SwitchAVSession“ im AVSession-Manager wird dieser darüber informiert, dass nun eine andere Medien-Wiedergabe gesteuert werden soll, falls ein solche selektiert wurde.

Methode: treeViewRenderer_GotFocus Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird aufgerufen, sobald die Baum-Ansicht „treeViewRenderer“ den Eingabefokus erhält. Sie macht die Hintergrund-Markierung durch die Methode „treeViewRenderer_LostFocus“ rückgängig.

Methode: treeViewRenderer_LostFocus Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird aufgerufen, sobald die Baum-Ansicht „treeViewRenderer“ den Eingabefokus verliert. Durch den Aufruf dieser Methode wird die selektierte AVSession mit einem grauen Hintergrund versehen, um trotz des verlorenen Eingabefokus dem User deutlich zu machen zu können, auf welche AVSession sich die angezeigten Informationen oder aber die Player-Steuerung beziehen.

Methode: UpdateAVSessionInfo Sichtbarkeit: private

Eingabe: UPnPDevice device, AVSession avSession, CommonStateInfo stateInfo

Ausgabe: void

Diese Methode wird durch das Event „OnActiveAVSessionInfo“ im AVSession-Manager aufgerufen und übergibt an die GUI die aktive AVSession in Form einer „AVSession“-Instanz, den Renderer, auf dem die Medien-Wiedergabe stattfindet, und die neuen Informationen in Form von „CommonStateInfo“-Instanzen. Die „avSession“ entspricht der zur Zeit steuerbaren Medien-Wiedergabe. Die Ausführung wird an den Thread delegiert, der das Control „listViewInfo“ erzeugt hat.

Methode: UpdateAVSessionInfoSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs args

Ausgabe: void

Diese Methode wird dafür verwendet, die in der „ListView“-Instanz „listViewInfo“ angezeigten Informationen zu aktualisieren, und sie übernimmt die Aktualisierung des Players und der dazugehörigen GUI-Komponenten.

Methode: UpdateBrowsingStructure Sichtbarkeit: private

Eingabe: CDSObject cdsObject, int index, int guiID, Control control

Ausgabe: void

Die Methode wird durch das Event „OnSingleCDSObjectSorted“ aufgerufen und wird dafür verwendet das durch „cdsObject“ abstrahierte Medien-Objekt in der GUI anzuzeigen. Der „index“ beschreibt dabei die exakte Position innerhalb der Medien-Liste, an der das Medium anzuzeigen ist. „guiID“ und „control“ beschreiben jeweils die GUI-Ansicht bzw. -Komponente, die mit diesem Medien-Objekt verknüpft sind. Die Ausführung wird an den Thread delegiert, der das Control „control“ erzeugt hat.

Methode: UpdateBrowsingStructureSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs args

Ausgabe: void

Diese Methode fügt das Medium an die exakte Position innerhalb der Medien-Liste in der GUI ein und erzeugt ein passendes Icon je nach Medien-Typ.

Methode: UpdateControl Sichtbarkeit: private

Eingabe: AVSession avSession

Ausgabe: void

Diese Methode wird verwendet, um der GUI zu signalisieren, dass der Player, der zur Steuerung der aktiven AVSession konfiguriert ist, aktualisiert bzw. zurückgesetzt werden müssen. Der Parameter „avSession“ beschreibt die betroffene Medien-Wiedergabe. Die Ausführung wird an den Thread delegiert, der das Control „treeViewRenderer“ erzeugt hat.

Methode: UpdateControlSink Sichtbarkeit: private

Eingabe: Object obj, EventArgs e

Ausgabe: void

Innerhalb dieser Methode erfolgt – falls notwendig – ein Reset des Players sowie die Erzeugung einer neuen „ListView“-Instanz, die Informationen über die neue AVSession aufnehmen kann.

Klasse: PropertyForm Eine Instanz dieser Klasse bietet dem Nutzer alle zur Verfügung stehenden Meta-Informationen zu einem Medien-Container oder einem Medien-Objekt. Durch einen modalen Dialog werden dem Nutzer neben den reinen Meta-Informationen auch Transfer-Protokoll-Informationen zur Verfügung gestellt.

Methode: listViewSelect_ItemActivate Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Diese Methode wird durch eine Aktivierung des gewünschten Menüpunktes aufgerufen und zeigt dem Nutzer die Meta- oder Transfer-Protokoll-Informationen.

Methode: PropertyForm Sichtbarkeit: public

Eingabe: MainForm parentForm, CDSObject cdsObject

Ausgabe: void

Hierbei handelt es sich um einen Konstruktor der Klasse „PropertyForm“. Übergeben wird eine Referenz auf das Hauptfenster und die durch den Klick-selektierte CDSObject-Instanz. Neben den GUI-Komponenten werden die Variablen „parentForm“ und „cds-Object“ initialisiert.

Methode: ShowProperty Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Diese Methode ermittelt per Reflection die Meta- sowie Transfer-Protokoll-Informationen zu dem Medien-Objekt.

Methode: ShowProtocolInfo Sichtbarkeit: private

Eingabe: int index

Ausgabe: void

Durch einen Klick auf die gewünschte Protokoll-Informationen, referenziert durch einen internen Index, werden die dazugehörigen Informationen wie „ContentURI“, „ImportURI“ und „ProtocolInfo“ ausgegeben.

Klasse: PropertyFormDevice Eine Instanz dieser Klasse zeigt die zu einem selektierten Server gehörigen Meta-Informationen an. Da sich die grundlegende Funktionsweise von einer Instanz der Klasse „PropertyForm“ kaum unterscheidet, könnte darüber nachgedacht werden, beide GUI-Komponenten zusammenzulegen. Dies wird jedoch auf zukünftige Versionen verschoben.

Methode: PropertyFormDevice Sichtbarkeit: public

Eingabe: MainForm parentForm, UPnPDevice device

Ausgabe: void

Hierbei handelt es sich um einen Konstruktor der Klasse „PropertyFormDevice“. Übergeben wird eine Referenz auf das Hauptfenster und den durch den Klick selektierte Server in Form einer UPnPDevice-Instanz. Neben den GUI-Komponenten werden die Variablen „parentForm“ und „device“ initialisiert, sowie einige Label-Informationen gesetzt.

Methode: ShowProperty Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Diese Methode ermittelt per Reflection die Meta-Informationen zu dem selektierten Server in Form eines UPnPDevice-Objekts. Da einige Informationen nur der internen Verwaltung dienen, werden diese durch entsprechende Kontrollausdrücke ausgeblendet.

Klasse: SelectRendererForm Eine Instanz dieser Klasse wird verwendet, wenn ein Medium abgespielt werden soll und zunächst noch kein Renderer ausgewählt wurde, so erscheint ein modaler Dialog, der den Nutzer auffordert einen Renderer anzugeben.

Methode: listViewSelectRenderer_ItemActivate Sichtbarkeit: private

Eingabe: object sender, EventArgs e

Ausgabe: void

Wurde ein Renderer ausgewählt, so kann per „EstablishNewAVSession“ im AVSession-Manager ein neuer Abspielvorgang initiiert werden.

Methode: RendererAdded Sichtbarkeit: public

Eingabe: UPnPDevice device

Ausgabe: void

Diese Methode leitet das Ereignis, dass ein neuer Renderer verfügbar ist, an die GUI weiter.

Methode: `RendererAddedSink` Sichtbarkeit: private

Eingabe: Object obj, EventArgs args

Ausgabe: void

Diese Methode wird aus der Methode „`RendererAdded`“ gerufen und veranlasst eine Prüfung, ob der neu verfügbare Renderer zu dem aktuell ausgewählten Medienobjekt kompatibel ist. Ist dies der Fall, so wird er in der GUI angezeigt, ansonsten nicht.

Methode: `SelectRendererForm` Sichtbarkeit: public

Eingabe: CDSObject cdsObject, MainForm calledFromForm

Ausgabe: void

Hierbei handelt es sich um einen Konstruktor der Klasse „`SelectRendererForm`“, wobei eine „`CDSObject`“-Instanz und eine Referenz auf das Hauptfenster übergeben wird. Neben der Initialisierung der GUI, wird die Methode „`ShowRenderers`“ aufgerufen.

Methode: `ShowRenderers` Sichtbarkeit: private

Eingabe: void

Ausgabe: void

Diese Methode fordert eine Liste verfügbarer Renderer an, testet sie auf Kompatibilität, und zeigt sie im positiven Fall in der GUI an.

1.1.2 Namespace: `UniDo.PG483.UPnP`

Klasse: `AVTransport` Diese Klasse wird verwendet, um eine vollständige, UPnP-AV-konforme Implementierung sicherzustellen. Basierend auf dem durch den Intel-Device-Builder ? erzeugten Quellcode, wurden einige Modifikationen vorgenommen, um multiple Media-Instanzen auf einem Gerät unterstützen zu können, wobei nur eine Instanz dieser Klasse pro Device notwendig ist. Bietet ein Device einen „`AVTransport`“-Dienst an, so wird automatisch ein Objekt dieser Klasse erzeugt, mit dessen Hilfe ein Zugriff auf die zahlreichen UPnP-AV-Aktionen dieses Dienstes möglich ist. Ein Objekt dieser Klasse wird verwendet, sobald ein UPnP-AV-Server oder ein UPnP-AV-Renderer im Netzwerk erscheint, das einen `AVTransport-Service` anbietet. Intern verwaltet es alle Instanz-bezogenen Informationen in einer verschachtelten Hash-Tabelle namens „`AVTransportInstances`“. Als Key wird die `AVTransportID` verwendet, dessen zugeordneter Wert wiederum einer Hash-Tabelle entspricht. Diese Hash-Tabelle erlaubt einen Zugriff auf die UPnP-Status-Variablen. Als Key wird der Name der Variablen verwendet, wobei der dazugehörige Wert einer „`UPnPStateVariable`“-Instanz entspricht.

Event: `OnGetPositionInfoStateInfo` Eingabe: AVTransport sender, CommonStateInfo stateInfo

Ausgabe: void

Dieses Event wird beim Aufruf der Methode „`Sink_GetPositionInfo`“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „`GetPositionInfo`“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „`Sink_GetPositionInfo`“ ausgeführt. Die erhaltenen Informationen (siehe UPnP-AV-

Spezifikation) werden zu je einer CommonStateInfo-Instanz zusammengefasst und weitergeleitet. Angemeldet ist die Methode „NewAVSessionInfoReceived“ im AVSessionManager.

Event: OnGetTransportInfoStateInfo Eingabe: AVTransport sender, CommonStateInfo stateInfo
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetTransportInfoStateInfo“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „GetTransportInfoStateInfo“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_GetTransportInfoStateInfo“ ausgeführt. Die erhaltenen Informationen (siehe UPnP-AV-Spezifikation) werden zu je einer CommonStateInfo-Instanz zusammengefasst und weitergeleitet. Angemeldet ist die Methode „NewAVSessionInfoReceived“ im AVSessionManager.

Event: OnResult_GetCurrentTransportActions Eingabe: AVTransport sender, UInt32 InstanceID, String Actions, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_GetCurrentTransportActions“ und „Error_Sink_GetCurrentTransportActions“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „GetCurrentTransportActions“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_GetCurrentTransportActions“ ausgeführt. Diese Event wird zur Zeit nicht verwendet. Es sollte in einer zukünftigen Implementierung durch ein passendes Event „OnGetCurrentTransportActions“ ersetzt werden, das die Informationen als CommonStateInfo-Instanz weiterleitet.

Event: OnResult_GetDeviceCapabilities Eingabe: AVTransport sender, UInt32 InstanceID, String PlayMedia, String RecMedia, String RecQualityModes, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_GetDeviceCapabilities“ und „Error_Sink_GetDeviceCapabilities“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „GetDeviceCapabilities“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_GetDeviceCapabilities“ aufgerufen. Dieses Event wird zur Zeit nicht verwendet. Es sollte in einer zukünftigen Implementierung durch ein passendes Event „OnGetDeviceCapabilities“ ersetzt werden, das die Informationen als CommonStateInfo-Instanz weiterleitet.

Event: OnResult_GetMediaInfo Eingabe: AVTransport sender, UInt32 InstanceID, UInt32 NrTracks, String MediaDuration, String CurrentURI, String CurrentURIMetaData, String NextURI, String NextURIMetaData, Enum_PlaybackStorageMedium PlaybackMedium, Enum_RecordStorageMedium RecordMedium, Enum_RecordMediumWriteStatus

WriteStatus, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_GetMediaInfo“ und „Error_Sink_GetMediaInfo“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „GetMediaInfo“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_GetMediaInfo“ aufgerufen. Dieses Event wird zur Zeit nicht verwendet. Es sollte in einer zukünftigen Implementierung durch ein passendes Event „OnGetMediaInfo“ ersetzt werden, das die Informationen als CommonStateInfo-Instanz weiterleitet.

Event: OnResult_GetPositionInfo Eingabe: AVTransport sender, UInt32 InstanceID, UInt32 Track, String TrackDuration, String TrackMetaData, String TrackURI, String RelTime, String AbsTime, Int32 RelCount, Int32 AbsCount, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_GetPositionInfo“ und „Error_Sink_GetPositionInfo“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „GetPositionInfo“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_GetPositionInfo“ aufgerufen. Dieses Event wird zur Zeit nicht verwendet. Es sollte in einer zukünftigen Implementierung durch ein passendes Event „OnGetPositionInfo“ ersetzt werden, das die Informationen als CommonStateInfo-Instanz weiterleitet.

Event: OnResult_GetTransportInfo Eingabe: AVTransport sender, UInt32 InstanceID, Enum_TransportState CurrentTransportState, Enum_TransportStatus CurrentTransportStatus, Enum_TransportPlaySpeed CurrentSpeed, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_GetTransportInfo“ und „Error_Sink_GetTransportInfo“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „GetTransportInfo“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_GetTransportInfo“ aufgerufen. Dieses Event wird zur Zeit nicht verwendet. Es sollte in einer zukünftigen Implementierung durch ein passendes Event „OnGetTransportInfo“ ersetzt werden, das die Informationen als CommonStateInfo-Instanz weiterleitet.

Event: OnResult_GetTransportSettings Eingabe: AVTransport sender, UInt32 InstanceID, Enum_CurrentPlayMode PlayMode, Enum_CurrentRecordQualityMode RecQualityMode, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_GetTransportSettings“ und „Error_Sink_GetTransportSettings“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „GetTransportSettings“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_GetTransportSettings“

aufgerufen. Dieses Event wird zur Zeit nicht verwendet. Es sollte in einer zukünftigen Implementierung durch ein passendes Event „OnGetTransportSettings“ ersetzt werden, das die Informationen als CommonStateInfo-Instanz weiterleitet.

Event: OnResult_Next Eingabe: AVTransport sender, UInt32 InstanceID, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_Next“ und „Error_Sink_Next“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „Next“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_Next“ aufgerufen. Dieses Event wird zur Zeit nicht verwendet.

Event: OnResult_Pause Eingabe: AVTransport sender, UInt32 InstanceID, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_Pause“ und „Error_Sink_Pause“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „Pause“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_Pause“ aufgerufen. Dieses Event wird zur Zeit nicht verwendet.

Event: OnResult_Play Eingabe: AVTransport sender, UInt32 InstanceID, Enum.TransportPlaySpeed Speed, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_Play“ und „Error_Sink_Play“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „Play“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_Play“ aufgerufen. Dieses Event wird zur Zeit nicht verwendet. Es sollte in einer zukünftigen Implementierung durch ein passendes Event „OnPlay“ ersetzt werden, das die Informationen als CommonStateInfo-Instanz weiterleitet.

Event: OnResult_Previous Eingabe: AVTransport sender, UInt32 InstanceID, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_Previous“ und „Error_Sink_Previous“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „Previous“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_Previous“ aufgerufen. Dieses Event wird zur Zeit nicht verwendet.

Event: OnResult_Seek Eingabe: AVTransport sender, UInt32 InstanceID, Enum.A_ARG_TYPE_SeekMode Unit, String Target, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink.Seek” und „Error.Sink.Seek” geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „Seek” erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink.Seek” aufgerufen. Dieses Event wird zur Zeit nicht verwendet. Es sollte in einer zukünftigen Implementierung durch ein passendes Event „OnSeek” ersetzt werden, das die Informationen als CommonStateInfo-Instanz weiterleitet.

Event: OnResult_SetAVTransportURI Eingabe: AVTransport sender, UInt32 InstanceID, String CurrentURI, String CurrentURIMetaData, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink.SetAVTransportURI” und „Error.Sink.SetAVTransportURI” geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „SetAVTransportURI” erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink.SetAVTransportURI” aufgerufen. Dieses Event wird zur Zeit nicht verwendet. Es sollte in einer zukünftigen Implementierung durch ein passendes Event „OnSetAVTransportURI” ersetzt werden, das die Informationen als CommonStateInfo-Instanz weiterleitet.

Event: OnResult_SetPlayMode Eingabe: AVTransport sender, UInt32 InstanceID, Enum.CurrentPlayMode NewPlayMode, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink.SetPlayMode” und „Error.Sink.SetPlayMode” geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „SetPlayMode” erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink.SetPlayMode” aufgerufen. Dieses Event wird zur Zeit nicht verwendet. Es sollte in einer zukünftigen Implementierung durch ein passendes Event „OnSetPlayMode” ersetzt werden, das die Informationen als CommonStateInfo-Instanz weiterleitet.

Event: OnResult_Stop Eingabe: AVTransport sender, UInt32 InstanceID, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink.Stop” und „Error.Sink.Stop” geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „Stop” erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink.Stop” aufgerufen. Dieses Event wird zur Zeit nicht verwendet. Es sollte in einer zukünftigen Implementierung durch ein passendes Event „OnStop” ersetzt werden, das die Informationen als CommonStateInfo-Instanz weiterleitet.

Event: StateVariable_LastChange Eingabe: AVTransport sender, CommonStateInfo stateInfo

Ausgabe: void

Dieses Event leitet per „CommonStateInfo“-Instanz die neu eingetroffene LastChange-Notifikation an den AVSessionManager weiter. Gerufen wird die Methode „LastChangesAVTransport“, die sich in der AVSessionManager-Instanz befindet.

Event: OnSubscribe Eingabe: AVTransport sender, bool Success

Ausgabe: void

Dieses Event wird geworfen, sobald eine Registrierung am Rendering-Control-Service des entsprechenden Devices durchgeführt wurde. Das Flag „Success“ signalisiert einen Erfolg oder aber einen Misserfolg.

Methode: _subscribe Sichtbarkeit: public

Eingabe: int Timeout

Ausgabe: void

Mit Hilfe dieser Methode erfolgt eine Subscription für den Dienst „AVTransport“. Für weitere Details bzgl. des Subscription-Mechanismus wird auf die UPnP-AV-Spezifikation verwiesen.

Methode: _subscribe_sink Sichtbarkeit: protected

Eingabe: UPnPService sender, bool OK

Ausgabe: void

Diese Methode wird als Callback bei einer erfolgten Subscription aufgerufen und leitet die eingegangene Information mit Hilfe des Events „OnSubscribe“ weiter.

Methode: AVTransport Sichtbarkeit: public

Eingabe: UPnPService s

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor der Klasse „AVTransport“. Übergeben wird eine UPnPService-Instanz, die mit diesem Objekt verknüpft wird.

Methode: Dispose Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Mit Hilfe dieser Methode können „Aufräum“-Operationen durchgeführt werden. Es erfolgt eine Abmeldung von den zahlreichen Events, sowie eine offizielle UPnP-konforme Un-Subscription-Mitteilung an das betroffene Gerät, um zu signalisieren, dass zukünftige Änderungen des Geräte-Zustandes nicht mehr an diesen Kontrollpunkt propagiert werden sollen.

Methode: Enum_A_ARG_TYPE_SeekMode_ToString Sichtbarkeit: public

Eingabe: Enum_A_ARG_TYPE_SeekMode en

Ausgabe: string

Diese Methode wandelt einen vorliegenden Enumerations-Wert „en“ in einen entsprechenden String um.

Methode: Enum_CurrentPlayMode_ToString Sichtbarkeit: public

Eingabe: Enum_CurrentPlayMode en

Ausgabe: string

Diese Methode wandelt einen vorliegenden Enumerations-Wert „en“ in einen entsprechenden String um.

Methode: Enum_CurrentRecordQualityMode_ToString Sichtbarkeit: public

Eingabe: Enum_CurrentRecordQualityMode en

Ausgabe: string

Diese Methode wandelt einen vorliegenden Enumerations-Wert „en“ in einen entsprechenden String um.

Methode: Enum_PlaybackStorageMedium_ToString Sichtbarkeit: public

Eingabe: Enum_PlaybackStorageMedium en

Ausgabe: string

Diese Methode wandelt einen vorliegenden Enumerations-Wert „en“ in einen entsprechenden String um.

Methode: Enum_RecordMediumWriteStatus_ToString Sichtbarkeit: public

Eingabe: Enum_RecordMediumWriteStatus en

Ausgabe: string

Diese Methode wandelt einen vorliegenden Enumerations-Wert „en“ in einen entsprechenden String um.

Methode: Enum_RecordStorageMedium_ToString Sichtbarkeit: public

Eingabe: Enum_RecordStorageMedium en

Ausgabe: string

Diese Methode wandelt einen vorliegenden Enumerations-Wert „en“ in einen entsprechenden String um.

Methode: Enum_TransportPlaySpeed_ToString Sichtbarkeit: public

Eingabe: Enum_TransportPlaySpeed en

Ausgabe: string

Diese Methode wandelt einen vorliegenden Enumerations-Wert „en“ in einen entsprechenden String um.

Methode: Enum_TransportState_ToString Sichtbarkeit: public

Eingabe: Enum_TransportState en

Ausgabe: string

Diese Methode wandelt einen vorliegenden Enumerations-Wert „en“ in einen entsprechenden String um.

Methode: Enum_TransportStatus_ToString Sichtbarkeit: public

Eingabe: Enum_TransportStatus en

Ausgabe: string

Diese Methode wandelt einen vorliegenden Enumerations-Wert „en“ in einen entsprechenden String um.

Methode: Error_Sink_GetCurrentTransportActions Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetCurrentTransportActions“ zu einem Fehler gekommen sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetDeviceCapabilities Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetDeviceCapabilities“ zu einem Fehler gekommen sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetMediaInfo Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetMediaInfo“ zu einem Fehler gekommen sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetPositionInfo Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetPositionInfo“ zu einem Fehler gekommen sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetTransportInfo Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetTransportInfo“ zu einem Fehler gekommen

sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetTransportSettings Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetTransportSettings“ zu einem Fehler gekommen sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_Next Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „Next“ zu einem Fehler gekommen sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_Pause Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „Pause“ zu einem Fehler gekommen sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_Play Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „Play“ zu einem Fehler gekommen sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_Previous Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „Previous“ zu einem Fehler gekommen sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_Seek Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „Seek“ zu einem Fehler gekommen sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetAVTransportURI Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetAVTransportURI“ zu einem Fehler gekommen sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetPlayMode Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetPlayMode“ zu einem Fehler gekommen sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_Stop Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „Stop“ zu einem Fehler gekommen sein, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: GetA_ARG_TYPE_InstanceID Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: UInt32

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „A_ARG_TYPE_InstanceID“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetA_ARG_TYPE_SeekMode Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „A_ARG_TYPE_SeekMode“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetA_ARG_TYPE_SeekTarget Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „A_ARG_TYPE_SeekTarget“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetAbsoluteTimePosition Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „AbsoluteTimePosition“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetAVTransportIDs Sichtbarkeit: public

Eingabe: void

Ausgabe: ArrayList

Diese Methode erzeugt eine „ArrayList“ aller bekannter AVTransportIDs auf dem entsprechenden Device, das durch die Instanz dieser Klasse („AVTransport“) eindeutig festgelegt ist.

Methode: GetAVTransportURI Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „AVTransportURI“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetAVTransportURIMetaData Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „AVTransportURIMetaData“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetCurrentPlayMode Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: Enum_CurrentPlayMode

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „CurrentPlayMode“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetCurrentRecordQualityMode Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: Enum_CurrentRecordQualityMode

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „CurrentRecordQualityMode“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetCurrentTrack Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: UInt32

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „CurrentTrack“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetCurrentTrackDuration Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „CurrentTrackDuration“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetCurrentTrackMetaData Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „CurrentTrackMetaData“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetCurrentTrackURI Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „CurrentTrackUR“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetCurrentTransportActions Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „CurrentTransportActions“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetCurrentTransportActions Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „GetCurrentTransportActions“.

Methode: GetCurrentTransportActions Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetCurrentTransportActions_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetCurrentTransportActions“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: GetDeviceCapabilities Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „GetDeviceCapabilities“.

Methode: GetDeviceCapabilities Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetDeviceCapabilities_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetDeviceCapabilities“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: GetMediaInfo Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „GetMediaInfo“.

Methode: GetMediaInfo Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetMediaInfo_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetMediaInfo“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: GetNextAVTransportURI Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „NextAVTransportURI“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetNextAVTransportURIMetaData Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „NextAVTransportURIMetaData“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetNumberOfTracks Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: UInt32

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „NumberOfTracks“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetPlaybackStorageMedium Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: Enum_PlaybackStorageMedium

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „PlaybackStorageMedium“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetPositionInfo Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „GetPositionInfo“.

Methode: GetPositionInfo Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetPositionInfo _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetPositionInfo“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: GetPossiblePlaybackStorageMedia Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „PossiblePlaybackStorageMedia“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetPossibleRecordStorageMedia Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „PossibleRecordStorageMedia“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetRecordMediumWriteStatus Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: Enum_RecordMediumWriteStatus

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „RecordMediumWriteStatus“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetRecordStorageMedium Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: Enum_RecordStorageMedium

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „RecordStorageMedium“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetRelativeTimePosition Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „RelativeTimePosition“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetStateVariable Sichtbarkeit: private

Eingabe: Int32 avTransportID, String VarName

Ausgabe: Object

Diese Methode stellt eine Hilfsstruktur dar. Mit ihrer Hilfe erfolgt ein Zugriff auf den Wert einer UPnP-Status-Variablen, wobei „avTransportID“ die AVTransportID bezeichnet und „VarName“ den Namen der UPnP-Status-Variablen. Als Rückgabe erhält man den Wert der AVTransport-spezifischen UPnP-Status-Variablen.

Methode: GetStateVariables Sichtbarkeit: private

Eingabe: Int32 avTransportID

Ausgabe: ArrayList

Diese Methode liefert alle zu einer AVTransport-Instanz bekannten UPnP-Status-Variablen als „ArrayList“, bestehend aus „UPnPStateVariable“-Instanzen.

Methode: GetTransportInfo Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „GetTransportInfo“.

Methode: GetTransportInfo Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetTransportInfo _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetTransportInfo“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: GetTransportPlaySpeed Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: Enum_TransportPlaySpeed

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „TransportPlaySpeed“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetTransportSettings Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „GetTransportSettings“.

Methode: GetTransportSettings Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetTransportSettings _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetTransportSettings“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: GetTransportState Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: Enum.TransportState

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „Transport-
State“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetTransportStatus Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: Enum.TransportState

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „Transport-
Status“ zu der durch „avTransportID“ angegebenen AVTransportID.

Methode: GetUnspecifiedValue Sichtbarkeit: private

Eingabe: string EnumType

Ausgabe: string

Methode: GetUPnPService Sichtbarkeit: public

Eingabe: void

Ausgabe: UPnPService

Mit Hilfe dieser Methode besteht die Möglichkeit, auf die durch ein Objekt dieser Klasse
gekapselte UPnPService-Instanz zuzugreifen.

Methode: LastChange_ModifiedSink Sichtbarkeit: protected

Eingabe: UPnPStateVariable Var, object NewValue

Ausgabe: void

Diese Methode wird aufgerufen, sobald eine Last-Change-Notifikation empfangen wurde.
Durch den Aufruf der Methode „ParseLastChange“ wird die durch die Variable „New-
Value“ übergebene Last-Change-XML-Nachricht geparkt und die enthaltenen neuen
Variablen-Belegungen werden intern gesetzt.

Methode: Next Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „Next“.

Methode: Next Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate.OnResult_Next _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „Next“ auf
dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: ParseLastChange Sichtbarkeit: private

Eingabe: string lastChangeValue

Ausgabe: void

Innerhalb dieser Methode erfolgt der eigentliche Parse-Vorgang mit Hilfe eines DOM-Parsers. An dieser Stelle könnte aus Optimierungsgründen in einer späteren Version auf einen SAX-Parser zurückgegriffen werden. Zunächst wird die betroffene AVTransport-Instanz ermittelt, indem die AVTransportID aus der Nachricht extrahiert wird. Nun kann die dieser AVTransportID-zugeordneten Hash-Tabelle ermittelt werden, die die Variablen dieser Instanz beinhaltet. Durch den Aufruf der Methode „SetStateVariables“ werden die betroffenen UPnP-Status-Variablen intern aktualisiert.

Methode: Pause Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „Pause“.

Methode: Pause Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_Pause _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „Pause“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: Play Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_TransportPlaySpeed Speed

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „Play“.

Methode: Play Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_Play _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „Play“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: Previous Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „Previous“.

Methode: Previous Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_Previous _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „Previous“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: Seek Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_SeekMode Unit, String Target

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „Seek“.

Methode: Seek Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_SeekMode Unit, String Target, object _Tag, Delegate_OnResult_Seek _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „Seek“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: SetAVTransportID Sichtbarkeit: public

Eingabe: Int32 avTransportID

Ausgabe: void

Mit Hilfe dieser Methode wird eine neue Hash-Tabelle mit den durch die gekapselten UPnP-Service-Instanz betroffenen UPnP-Status-Variablen erzeugt. Durch den Aufruf von „GetMediaInfo“, „GetTransportInfo“, „GetTransportSettings“ und „GetPositionInfo“ können alle diese Verbindung betreffenden Informationen mit Hilfe von SOAP-Nachrichten abgefragt und intern gesetzt werden.

Methode: SetAVTransportURI Sichtbarkeit: public

Eingabe: UInt32 InstanceID, String CurrentURI, String CurrentURIMetaData

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetAVTransportURI“.

Methode: SetAVTransportURI Sichtbarkeit: public

Eingabe: UInt32 InstanceID, String CurrentURI, String CurrentURIMetaData, object _Tag, Delegate_OnResult_SetAVTransportURI _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetAVTransportURI“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: SetPlayMode Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_CurrentPlayMode NewPlayMode

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetPlayMode“.

Methode: SetPlayMode Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_CurrentPlayMode NewPlayMode, object _Tag, Delegate_OnResult_SetPlayMode _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetPlayMode“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: SetRelativeTimePosition Sichtbarkeit: public

Eingabe: Int32 avTransportID, String value

Ausgabe: void

Unter Anwendung dieser Methode wird die relative Abspielzeit innerhalb einer durch „avTransportID“ referenzierten Medienwiedergabe auf den gewünschten Wert „value“ gesetzt.

Methode: SetStateVariables Sichtbarkeit: private

Eingabe: Hashtable stateVariables, int instanceID, XmlNode node

Ausgabe: void

Diese Methode erhält als Eingabe die von der Änderung betroffene Hash-Tabelle, die entsprechende AVTransportID und eine XmlNode-Instanz, die die Änderungen enthält. Wurde eine Variable erfolgreich aktualisiert, wird mit Hilfe des Events „OnStateVariable_LastChange“ ein Ereignis ausgelöst, das die AVTransport-Instanz und eine CommonStateInfo-Instanz übergeben bekommt. Dieses CommonStateInfo-Objekt stellt ein 3er-Tupel bestehend aus dem Variablen-Namen, dem neuen Wert und der AVTransportID. Als Sonderfall kann das XML-Element „CurrentTrackEmbeddedMetaData“ betrachtet werden, das zunächst nicht weiter behandelt wird. Die UPnP-Spezifikation sieht solche eingebetteten Informationen nicht vor. Intel scheint die Problematik bzgl. der Meta-Information bereits erkannt zu haben und führt ein schnelles Workaround durch. Allerdings wird von keinem Intel-Renderer ? diese Information sinnvoll gesetzt, so dass auf einen expliziten Parse-Vorgang verzichtet wurde.

Methode: SetUnspecifiedValue Sichtbarkeit: public

Eingabe: string EnumType, string val

Ausgabe: void

Methode: Sink_GetCurrentTransportActions Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetCurrentTransportActions“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetCurrentTransportActions“ diese Antwort wei-

tergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetDeviceCapabilities Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetDeviceCapabilities“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetDeviceCapabilities“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetMediaInfo Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetMediaInfo“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetMediaInfo“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetPositionInfo Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetPositionInfo“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetPositionInfo“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetTransportInfo Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetTransportInfo“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetTransportInfo“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetTransportSettings Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetTransportSettings“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetTransportSettings“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_Next Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „Next“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_Next“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_Pause Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „Pause“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_Pause“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_Play Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „Play“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_Play“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_Previous Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „Previous“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event

„Delegate_OnResult_Previous“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_Seek Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „Seek“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_Seek“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetAVTransportURI Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetAVTransportURI“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetAVTransportURI“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetPlayMode Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetPlayMode“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetPlayMode“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_Stop Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „Stop“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_Stop“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Stop Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „Stop“.

Methode: Stop Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_Stop _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „Stop“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: Sync_GetCurrentTransportActions Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out String Actions

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetCurrentTransportActions“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist. Da die Variable „Actions“ als Referenz übergeben wird, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetDeviceCapabilities Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out String PlayMedia, out String RecMedia, out String RecQualityModes

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetDeviceCapabilities“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist. Da die Variablen „PlayMedia“, „RecMedia“ und „RecQualityModes“ als Referenz übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync_GetMediaInfo Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out UInt32 NrTracks, out String MediaDuration, out String CurrentURI, out String CurrentURIMetaData, out String NextURI, out String NextURIMetaData, out Enum_PlaybackStorageMedium PlayMedium, out Enum_RecordStorageMedium RecordMedium, out Enum_RecordMediumWriteStatus WriteStatus

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetMediaInfo“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist. Da die Variablen „NrTracks“, „MediaDuration“, „CurrentURI“, „CurrentURIMetaData“, „NextURI“, „NextURIMetaData“, „PlayMedium“, „RecordMedium“ und „WriteStatus“ als Referenz übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync_GetPositionInfo Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out UInt32 Track, out String TrackDuration, out String

TrackMetaData, out String TrackURI, out String RelTime, out String AbsTime, out Int32 RelCount, out Int32 AbsCount

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetPositionInfo“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist. Da die Variablen „Track“, „TrackDuration“, „TrackURI“, „CurrentURIMetaData“, „RelTime“, „AbsTime“, „RelCount“ und „AbsCount“ als Referenz übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync_GetTransportInfo Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out Enum_TransportState CurrentTransportState, out Enum_TransportStatus CurrentTransportStatus, out Enum_TransportPlaySpeed CurrentSpeed

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetTransportInfo“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist. Da die Variablen „CurrentTransportState“, „CurrentTransportStatus“ und „CurrentSpeed“ als Referenz übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync_GetTransportSettings Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out Enum_CurrentPlayMode PlayMode, out Enum_CurrentRecordQualityMode RecQualityMode

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetTransportSettings“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist. Da die Variablen „PlayMode“ und „RecQualityMode“ als Referenz übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync_Next Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „Next“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: Sync_Pause Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „Pause“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: Sync_Play Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „Play“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: Sync.Previous Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „Previous“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: Sync.Seek Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_SeekMode Unit, String Target

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „Seek“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: Sync.SetAVTransportURI Sichtbarkeit: public

Eingabe: UInt32 InstanceID, String CurrentURI, String CurrentURIMetaData

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetAVTransportURI“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: Sync.SetPlayMode Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_CurrentPlayMode NewPlayMode

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetPlayMode“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Methode: Sync.Stop Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „Stop“ auf dem Device, das dieser „AVTransport“-Instanz zugeordnet ist.

Property: HasAction_GetCurrentTransportActions Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetCurrentTransportActions“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasAction_GetDeviceCapabilities Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetDeviceCapabilities“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasAction_GetMediaInfo Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetMediaInfo“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasAction_GetPositionInfo Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetPositionInfo“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasAction_GetTransportInfo Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetTransportInfo“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasAction_GetTransportSettings Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetTransportSettings“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasAction_Next Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „Next“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasAction_Pause Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Aktion „Pause“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasAction_Play Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „Play“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasAction_Previous Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „Previous“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasAction_Seek Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „Seek“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasAction_SetAVTransportURI Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetAVTransportURI“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasAction_SetPlayMode Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetPlayMode“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasAction_Stop Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Aktion „Stop“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_A_ARG_TYPE_InstanceID Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_InstanceID“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_A_ARG_TYPE_SeekMode Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_SeekMode“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_A_ARG_TYPE_SeekTarget Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_SeekTarget“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_AbsoluteCounterPosition Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „AbsoluteCounterPosition“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_AbsoluteTimePosition Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „AbsoluteTimePosition“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_AVTransportURI Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Status-Variable „AVTransportURI“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_AVTransportURIMetaData Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „AVTransportURIMetaData“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_CurrentMediaDuration Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „CurrentMediaDuration“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_CurrentPlayMode Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „CurrentPlayMode“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_CurrentRecordQualityMode Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „CurrentRecordQualityMode“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_CurrentTrack Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „CurrentTrack“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_CurrentTrackDuration Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Status-Variable „CurrentTrackDuration“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_CurrentTrackMetadata Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „CurrentTrackMetadata“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_CurrentTrackURI Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „CurrentTrackURI“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_CurrentTransportActions Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „CurrentTransportActions“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_LastChange Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „LastChange“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_NextAVTransportURI Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „NextAVTransportURI“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_NextAVTransportURIMetaData Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Status-Variable „NextAVTransportURIMetaData“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_NumberOfTracks Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „NumberOfTracks“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_PlaybackStorageMedium Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „PlaybackStorageMedium“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_PossiblePlaybackStorageMedia Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „PossiblePlaybackStorageMedia“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_PossibleRecordQualityModes Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „PossibleRecordQualityModes“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_PossibleRecordStorageMedia Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „PossibleRecordStorageMedia“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_RecordMediumWriteStatus Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „RecordMediumWriteStatus“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_RecordStorageMedium Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „RecordStorageMedium“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_RelativeCounterPosition Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „RelativeCounterPosition“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_RelativeTimePosition Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „RelativeTimePosition“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_TransportPlaySpeed Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „TransportPlaySpeed“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_TransportState Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieses Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „TransportState“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: HasStateVariable_TransportStatus Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Status-Variable „TransportStatus“ implementiert, die Teil des UPnP-Service „AVTransport“ ist.

Property: UPnPDevice Sichtbarkeit: public

Eingabe: void

Ausgabe: UPnPDevice

Dieses Property liefert das mit diesem Objekt verknüpfte Device in Form einer UPnPDevice-Instanz zurück.

Property: Values_A_ARG_TYPE_SeekMode Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Dieses Property liefert alle möglichen Werte, die eine Variablen vom Typ A_ARG_TYPE_SeekMode annehmen kann.

Property: Values_CurrentPlayMode Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Dieses Property liefert alle möglichen Werte, die die UPnP-Status-Variable „CurrentPlayMode“ annehmen kann.

Property: Values_CurrentRecordQualityMode Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Dieses Property liefert alle möglichen Werte, die die UPnP-Status-Variable „CurrentRecordQualityMode“ annehmen kann.

Property: Values_PlaybackStorageMedium Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Dieses Property liefert alle möglichen Werte, die die UPnP-Status-Variable „PlaybackStorageMedium“ annehmen kann.

Property: Values_RecordMediumWriteStatus Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Dieses Property liefert alle möglichen Werte, die die UPnP-Status-Variable „RecordMediumWriteStatus“ annehmen kann.

Property: Values_RecordStorageMedium Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Dieses Property liefert alle möglichen Werte, die die UPnP-Status-Variable „RecordStorageMedium“ annehmen kann.

Property: Values_TransportPlaySpeed Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Dieses Property liefert alle möglichen Werte, die die UPnP-Status-Variable „TransportPlaySpeed“ annehmen kann.

Property: Values_TransportState Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Dieses Property liefert alle möglichen Werte, die die UPnP-Status-Variable „TransportState“ annehmen kann.

Property: Values_TransportStatus Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Dieses Property liefert alle möglichen Werte, die die UPnP-Status-Variable „TransportStatus“ annehmen kann.

Property: VERSION Sichtbarkeit: public

Eingabe: void

Ausgabe: double

Dieses Property liefert die Versionsnummer des zu verwendenden Rendering-Services zurück.

Klasse: ConnectionManager Diese Klasse wird verwendet, um eine vollständige UPnP-AV-konforme Implementierung sicherzustellen. Basierend auf der durch den Intel-Device-Builder ? erzeugten Implementierung, mussten nur wenige Modifikationen vorgenommen werden. Bietet ein Device einen „ConnectionManager“-Dienst an, so wird automatisch ein Objekt dieser Klasse erzeugt mit dessen Hilfe ein Zugriff auf die zahlreichen UPnP-AV-Aktionen möglich ist. Ein Objekt dieser Klasse wird verwendet, sobald ein UPnP-AV-Server oder ein UPnP-AV-Renderer im Netzwerk erscheint, das einen ConnectionManager-Service anbietet. Mit Hilfe dieser objektorientierten Kapselung kann nun ein Zugriff auf die angebotenen UPnP-AV-Aktionen erfolgen.

Event: OnResult_ConnectionComplete Eingabe: ConnectionManager sender, Int32 ConnectionID, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_ConnectionComplete“ und „Error_Sink_ConnectionComplete“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „ConnectionComplete“ erfolgreich aufgerufen

wurde, wird bei Erhalt der Antwort die Methode „Sink_ConnectionComplete“ aufgerufen. Dieses Event wird zur Zeit nicht verwendet.

Event: OnResult_GetCurrentConnectionIDs Eingabe: ConnectionManager sender, String ConnectionIDs, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_GetCurrentConnectionIDs“ und „Error_Sink_GetCurrentConnectionIDs“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „GetCurrentConnectionIDs“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_ConnectionComplete“ aufgerufen. Dieses Event wird zur Zeit nicht verwendet.

Event: OnResult_GetCurrentConnectionInfo Eingabe: ConnectionManager sender, Int32 ConnectionID, Int32 RcsID, Int32 AVTransportID, String ProtocolInfo, String PeerConnectionManager, Int32 PeerConnectionID, Enum_A_ARG_TYPE_Direction Direction, Enum_A_ARG_TYPE_ConnectionStatus Status, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_GetCurrentConnectionInfo“ und „Error_Sink_GetCurrentConnectionInfo“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „GetCurrentConnectionInfo“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_GetCurrentConnectionInfo“ aufgerufen. Dieses Event wird zur Zeit nicht verwendet.

Event: OnResult_GetProtocolInfo Eingabe: ConnectionManager sender, String Source, String Sink, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_GetProtocolInfo“ und „Error_Sink_GetCurrentConnectionInfo“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „GetProtocolInfo“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_GetProtocolInfo“ aufgerufen. Aufgerufen wird durch dieses Event die Methode „GetProtocolInfoSink“ in der entsprechenden AVSession-Instanz.

Event: OnResult_PrepareForConnection Eingabe: ConnectionManager sender, String RemoteProtocolInfo, String PeerConnectionManager, Int32 PeerConnectionID, Enum_A_ARG_TYPE_Direction Direction, Int32 ConnectionID, Int32 AVTransportID, Int32 RcsID, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_PrepareForConnection“ und „Error_Sink_PrepareForConnection“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „PrepareForConnection“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_PrepareForConnection“

aufgerufen.

Event: OnStateVariable_CurrentConnectionIDs Eingabe: ConnectionManager sender, String NewValue

Ausgabe: void

Dieses Event wird geworfen, falls es eine Zustandsänderung bzgl. der offenen Verbindung auf dem durch ein Objekt dieser Klasse gekapselten Device ergeben hat. Eine solche Zustandsänderung tritt z. B. bei der Initiierung einer neuen Verbindung oder beim Schließen einer bereits existierenden Verbindung auf.

Event: OnStateVariable_SinkProtocolInfo Eingabe: ConnectionManager sender, String NewValue

Ausgabe: void

Dieses Event wird geworfen, falls eine Protokoll-Änderung auf dem durch ein Objekt dieser Klasse gekapselten Device registriert wurde. Die betroffene UPnP-Status-Variable heißt „SinkProtocolInfo“.

Event: OnStateVariable_SourceProtocolInfo Eingabe: void

Ausgabe: ConnectionManager sender, String NewValue

Dieses Event wird geworfen, falls eine Protokoll-Änderung auf dem durch ein Objekt dieser Klasse gekapselten Device registriert wurde. Die betroffene UPnP-Status-Variable heißt „SourceProtocolInfo“.

Event: OnSubscribe Eingabe: ConnectionManager sender, bool Success

Ausgabe: void

Dieses Event wird geworfen, sobald eine Registrierung am Rendering-Control-Service des entsprechenden Devices durchgeführt wurde. Das Flag „Success“ signalisiert einen Erfolg oder aber einen Misserfolg.

Methode: _subscribe Sichtbarkeit: public

Eingabe: int Timeout

Ausgabe: void

Mit Hilfe dieser Methode erfolgt eine Subscription für den Dienst „ConnectionManager“. Für weitere Details bzgl. des Subscription-Mechanismus wird auf die UPnP-AV-Spezifikation verwiesen.

Methode: _subscribe_sink Sichtbarkeit: protected

Eingabe: UPnPService sender, bool OK

Ausgabe: void

Diese Methode wird als Callback bei einer erfolgten Subscription aufgerufen.

Methode: ConnectionComplete Sichtbarkeit: public

Eingabe: Int32 ConnectionID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „ConnectionComplete“.

Methode: ConnectionComplete Sichtbarkeit: public

Eingabe: Int32 ConnectionID, object _Tag, Delegate_OnResult_ConnectionComplete _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „ConnectionComplete“ auf dem Device, das dieser „ConnectionManager“-Instanz zugeordnet ist.

Methode: ConnectionManager Sichtbarkeit: public

Eingabe: UPnPService s

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor der Klasse „ConnectionManager“. Übergeben wird eine UPnPService-Instanz, die mit diesem Objekt verknüpft wird.

Methode: CurrentConnectionIDs_ModifiedSink

Sichtbarkeit: protected

Eingabe: UPnPStateVariable Var, object NewValue

Ausgabe: void

Wurde eine Veränderung bzgl. der offenen Verbindungen an einem Device registriert, so wird die Methode per Event gerufen. Diese Methode wiederum leitet die neuen Informationen per Event „OnStateVariable.CurrentConnectionIDs“ an dort registrierte Methoden weiter. In der vorliegenden Implementierung ist allein die Methode „GetCurrentConnectionIDsRendererSink“ an diesem Event angemeldet.

Methode: Dispose Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Mit Hilfe dieser Methode können „Aufräum“-Operationen durchgeführt werden. Er erfolgt eine Abmeldung von den zahlreichen Events, sowie eine offizielle UPnP-konforme Un-Subscription-Mitteilung an das betroffene Gerät, um zu signalisieren, dass zukünftige Änderungen bzgl. des Geräte-Zustandes nicht mehr an diesen Kontrollpunkt propagiert werden sollen.

Methode: Enum_A_ARG_TYPE_Direction_ToString Sichtbarkeit: public

Eingabe: Enum_A_ARG_TYPE_Direction en

Ausgabe: string

Diese Methode wandelt einen vorliegenden Enumerations-Wert „en“ in einen entsprechenden String um.

Methode: Error_Sink_ConnectionComplete Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „ConnectionComplete“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetCurrentConnectionIDs Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetCurrentConnectionIDs“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetCurrentConnectionInfo Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetCurrentConnectionInfo“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetProtocolInfo Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetProtocolInfo“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_PrepareForConnection Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „PrepareForConnection“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: GetCurrentConnectionIDs Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „GetCurrentConnectionIDs“.

Methode: GetCurrentConnectionIDs Sichtbarkeit: public

Eingabe: object _Tag, Delegate_OnResult_GetCurrentConnectionIDs _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetCurrentConnectionIDs“ auf dem Device, das dieser „ConnectionManager“-Instanz zugeordnet ist.

Methode: GetCurrentConnectionInfo Sichtbarkeit: public

Eingabe: Int32 ConnectionID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „GetCurrentConnectionInfo“.

Methode: GetCurrentConnectionInfo Sichtbarkeit: public

Eingabe: Int32 ConnectionID, object _Tag, Delegate_OnResult_GetCurrentConnectionInfo _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetCurrentConnectionInfo“ auf dem Device, das dieser „ConnectionManager“-Instanz zugeordnet ist.

Methode: GetProtocolInfo Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „GetProtocolInfo“.

Methode: GetProtocolInfo Sichtbarkeit: public

Eingabe: object _Tag, Delegate_OnResult_GetCurrentConnectionInfo _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetProtocolInfo“ auf dem Device, das dieser „ConnectionManager“-Instanz zugeordnet ist.

Methode: GetUnspecifiedValue Sichtbarkeit: public

Eingabe: string EnumType

Ausgabe: string

Methode: GetUPnPService Sichtbarkeit: public

Eingabe: void

Ausgabe: UPnPService

Mit Hilfe dieser Methode besteht die Möglichkeit, auf die durch ein Objekt dieser Klasse gekapselte UPnPService-Instanz zuzugreifen.

Methode: PrepareForConnection Sichtbarkeit: public

Eingabe: String RemoteProtocolInfo, String PeerConnectionManager, Int32 PeerConnectionID, Enum_A_ARG_TYPE_Direction Direction

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „PrepareForConnection“.

Methode: PrepareForConnection Sichtbarkeit: public

Eingabe: String RemoteProtocolInfo, String PeerConnectionManager, Int32 PeerConnectionID, Enum_A_ARG_TYPE_Direction Direction, object _Tag,

Delegate_OnResult_PrepareForConnection _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „PrepareForConnection“ auf dem Device, das dieser „ConnectionManager“-Instanz zugeordnet ist.

Methode: SetUnspecifiedValue Sichtbarkeit: public

Eingabe: string EnumType, string val

Ausgabe: void

Methode: Sink_ConnectionComplete Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „ConnectionComplete“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_ConnectionComplete“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetCurrentConnectionIDs Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetCurrentConnectionIDs“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder

wird per Event „Delegate_OnResult_GetCurrentConnectionIDs“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetCurrentConnectionInfo Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetCurrentConnectionInfo“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetCurrentConnectionInfo“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetProtocolInfo Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetProtocolInfo“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetProtocolInfo“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_PrepareForConnection Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „PrepareForConnection“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_PrepareForConnection“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: SinkProtocolInfo_ModifiedSink Sichtbarkeit: protected

Eingabe: UPnPStateVariable Var, object NewValue

Ausgabe: void

Wurde eine Protokoll-Änderung an einem Device registriert, so wird die vorliegenden Methode per Event gerufen. Diese Methode wiederum leitet die neuen Informationen per Event „OnStateVariable_SinkProtocolInfo“ an dort registrierte Methoden weiter. In der vorliegenden Implementierung ist allein die Methode „GetSinkProtocolInfo“ an diesem Event angemeldet.

Methode: SourceProtocolInfo.ModifiedSink Sichtbarkeit: protected

Eingabe: UPnPStateVariable Var, object NewValue

Ausgabe: void

Wurde eine Protokoll-Änderung an einem Device registriert, so wird die vorliegenden Methode per Event gerufen. Diese Methode wiederum leitet die neuen Informationen per Event „OnStateVariable.SourceProtocolInfo“ an dort registrierte Methoden weiter. In der vorliegenden Implementierung ist allein die Methode „GetSourceProtocolInfo“ an diesem Event angemeldet.

Methode: Sync.ConnectionComplete Sichtbarkeit: public

Eingabe: Int32 ConnectionID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „ConnectionComplete“ auf dem Device, das dieser „ConnectionManager“-Instanz zugeordnet ist.

Methode: Sync.GetCurrentConnectionIDs Sichtbarkeit: public

Eingabe: out String ConnectionIDs

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetCurrentConnectionIDs“ auf dem Device, das dieser „ConnectionManager“-Instanz zugeordnet ist. Da die Variable „ConnectionIDs“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync.GetCurrentConnectionInfo Sichtbarkeit: public

Eingabe: Int32 ConnectionID, out Int32 RcsID, out Int32 AVTransportID, out String ProtocolInfo, out String PeerConnectionManager, out Int32 PeerConnectionID, out Enum_A_ARG_TYPE_Direction Direction, out Enum_A_ARG_TYPE_ConnectionStatus Status

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetCurrentConnectionInfo“ auf dem Device, das dieser „ConnectionManager“-Instanz zugeordnet ist. Da die Variablen „RcsID“, „AVTransportID“, „ProtocolInfo“, „PeerConnectionManager“, „PeerConnectionID“, „Direction“ und „Status“ als Referenz übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync.GetProtocolInfo Sichtbarkeit: public

Eingabe: out String Source, out String Sink

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetProtocolInfo“ auf dem Device, das dieser „ConnectionManager“-Instanz zugeordnet ist. Da die Variablen „Source“ und „Sink“ als Referenz übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync.PrepareForConnection Sichtbarkeit: public

Eingabe: String RemoteProtocolInfo, String PeerConnectionManager, Int32 PeerConnectionID, Enum_A_ARG_TYPE_Direction Direction, out Int32 ConnectionID, out Int32 AVTransportID, out Int32 RcsID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetProtocolInfo“ auf dem Device, das dieser „ConnectionManager“-Instanz zugeordnet ist. Da die Variablen „ConnectionID“, „AVTransportID“ und „RcsID“ als Referenz übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Property: A_ARG_TYPE_AVTransportID Sichtbarkeit: public

Eingabe: void

Ausgabe: Int32

Dieses Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A_ARG_TYPE_AVTransportID“ zurück.

Property: A_ARG_TYPE_ConnectionID Sichtbarkeit: public

Eingabe: void

Ausgabe: Int32

Dieses Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A_ARG_TYPE_ConnectionID“ zurück.

Property: A_ARG_TYPE_ConnectionManager Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Dieses Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A_ARG_TYPE_ConnectionManager“ zurück.

Property: A_ARG_TYPE_ConnectionStatus Sichtbarkeit: public

Eingabe: void

Ausgabe: Enum_A_ARG_TYPE_ConnectionStatus

Dieses Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A_ARG_TYPE_ConnectionStatus“ zurück.

Property: A_ARG_TYPE_Direction Sichtbarkeit: public

Eingabe: void

Ausgabe: Enum_A_ARG_TYPE_Direction

Dieses Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A_ARG_TYPE_Direction“ zurück.

Property: A_ARG_TYPE_ProtocolInfo Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Dieses Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A_ARG_TYPE_ProtocolInfo“ zurück.

Property: A_ARG_TYPE_RcsID Sichtbarkeit: public

Eingabe: void

Ausgabe: Int32

Dieses Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A_ARG_TYPE_RcsID“ zurück.

Property: CurrentConnectionIDs Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Dieses Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „CurrentConnectionIDs“ zurück.

Property: HasAction_ConnectionComplete Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „ConnectionComplete“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasAction_GetCurrentConnectionIDs Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetCurrentConnectionIDs“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasAction_GetCurrentConnectionInfo Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetCurrentConnectionInfo“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasAction_GetProtocolInfo Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetProtocolInfo“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasAction_PrepareForConnection Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „PrepareForConnection“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasStateVariable_A_ARG_TYPE_AVTransportID Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_AVTransportID“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasStateVariable_A_ARG_TYPE_ConnectionID Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_ConnectionID“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasStateVariable_A_ARG_TYPE_ConnectionManager Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_ConnectionManager“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasStateVariable_A_ARG_TYPE_ConnectionStatus Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_ConnectionStatus“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasStateVariable_A_ARG_TYPE_Direction Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_Direction“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasStateVariable_A_ARG_TYPE_ProtocolInfo Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_ProtocolInfo“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasStateVariable_A_ARG_TYPE_RcsID Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_RcsID“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasStateVariable_CurrentConnectionIDs Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „CurrentConnectionIDs“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasStateVariable_SinkProtocolInfo Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „SinkProtocolInfo“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist.

Property: HasStateVariable_SourceProtocolInfo Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „SourceProtocolInfo“ implementiert, die Teil des UPnP-Service „ConnectionManager“ ist. StateVariableModifiedHandler

Property: SourceProtocolInfo Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Dieses Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „SourceProtocolInfo“ zurück.

Property: UPnPDevice Sichtbarkeit: public

Eingabe: void

Ausgabe: UPnPDevice

Diese Property liefert das mit diesem Objekt verknüpfte Device in Form einer UPnPDevice-Instanz zurück.

Property: Values_A_ARG_TYPE_Direction Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Diese Property liefert alle möglichen Werte, die eine Variablen vom Typ A_ARG_TYPE_Direction annehmen kann.

Property: VERSION Sichtbarkeit: public

Eingabe: void

Ausgabe: double

Diese Property liefert die Versions-Nummer des zu verwendenden Rendering-Services zurück.

Klasse: ContentDirectory Diese Klasse wird verwendet, um eine vollständige, UPnP-AV-konforme Implementierung sicherzustellen. Basierend auf der durch den Intel-Device-Builder ? erzeugten Implementierung, mussten nur wenige Modifikationen vorgenommen werden. Bietet ein Device einen „ContentDirectory“-Dienst an, so wird automatisch ein Objekt dieser Klasse erzeugt mit dessen Hilfe ein Zugriff auf die zahlreichen UPnP-AV-Aktionen möglich ist. Ein Objekt dieser Klasse wird verwendet, sobald ein UPnP-AV-Server oder ein UPnP-AV-Renderer im Netzwerk erscheint, das einen ContentDirectory-Service anbietet. Mit Hilfe dieser objektorientierten Kapselung kann nun ein Zugriff auf die angebotenen UPnP-AV-Aktionen erfolgen.

Event: OnResult_Browse Eingabe: ContentDirectory sender, String ObjectID, Enum_A_ARG_TYPE_BrowseFlag BrowseFlag, String Filter, UInt32 StartingIndex, UInt32 RequestedCount, String SortCriteria, String Result, UInt32 NumberReturned, UInt32 TotalMatches, UInt32 UpdateID, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_Browse“ und „Error_Sink_Browse“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „Browse“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_Browse“ aufgerufen.

Event: OnResult_CreateObject Eingabe: ContentDirectory sender, String ContainerID, String Elements, String ObjectID, String Result, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_CreateObject“ und „Error_Sink_CreateObject“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „CreateObject“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_CreateObject“ aufgerufen.

Event: OnResult_CreateReference Eingabe: ContentDirectory sender, String ContainerID, String ObjectID, String NewID, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_CreateReference“ und „Error_Sink_CreateReference“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „CreateReference“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_CreateReference“ aufgerufen.

Event: OnResult_DeleteResource Eingabe: ContentDirectory sender, Uri ResourceURI, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_DeleteResource“ und „Error_Sink_DeleteResource“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „DeleteResource“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_DeleteResource“ aufgerufen.

Event: OnResult_DestroyObject Eingabe: ContentDirectory sender, String ObjectID, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_DestroyObject“ und „Error_Sink_DestroyObject“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „DestroyObject“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_DestroyObject“ aufgerufen.

Event: OnResult_ExportResource Eingabe: ContentDirectory sender, Uri SourceURI, Uri DestinationURI, UInt32 TransferID, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_ExportResource“ und „Error_Sink_ExportResource“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „ExportResource“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_ExportResource“ aufgerufen.

Event: OnResult_GetSearchCapabilities Eingabe: ContentDirectory sender, String SearchCaps, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_GetSearchCapabilities“ und „Error_Sink_GetSearchCapabilities“ geworfen, falls eine Methode sich für dieses Ereignis

registriert hat. Nachdem die UPnP-Aktion „GetSearchCapabilities“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink.GetSearchCapabilities“ aufgerufen.

Event: OnResult_GetSortCapabilities Eingabe: ContentDirectory sender, String SortCaps, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink.GetSortCapabilities“ und „Error_Sink.GetSortCapabilities“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „GetSortCapabilities“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink.GetSortCapabilities“ aufgerufen.

Event: OnResult_GetSystemUpdateID Eingabe: ContentDirectory sender, UInt32 Id, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink.GetSystemUpdateID“ und „Error_Sink.GetSystemUpdateID“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „GetSystemUpdateID“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink.GetSystemUpdateID“ aufgerufen.

Event: OnResult_GetTransferProgress Eingabe: ContentDirectory sender, UInt32 TransferID, Enum_A_ARG_TYPE_TransferStatus TransferStatus, String TransferLength, String TransferTotal, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink.GetTransferProgress“ und „Error_Sink.GetTransferProgress“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „GetTransferProgress“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink.GetTransferProgress“ aufgerufen.

Event: OnResult_ImportResource Eingabe: ContentDirectory sender, Uri SourceURI, Uri DestinationURI, UInt32 TransferID, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink.ImportResource“ und „Error_Sink.ImportResource“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „ImportResource“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink.ImportResource“ aufgerufen.

Event: OnResult_Search Eingabe: ContentDirectory sender, String ContainerID, String SearchCriteria, String Filter, UInt32 StartingIndex, UInt32 RequestedCount, String SortCriteria, String Result, UInt32 NumberReturned, UInt32 TotalMatches, UInt32 UpdateID, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_Search“ und „Error_Sink_Search“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „Search“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_Search“ aufgerufen.

Event: OnResult_StopTransferResource Eingabe: ContentDirectory sender, UInt32 TransferID, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_StopTransferResource“ und „Error_Sink_StopTransferResource“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „StopTransferResource“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_StopTransferResource“ aufgerufen.

Event: OnResult_UpdateObject Eingabe: ContentDirectory sender, String ObjectID, String CurrentTagValue, String NewTagValue, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methoden „Sink_UpdateObject“ und „Error_Sink_UpdateObject“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat. Nachdem die UPnP-Aktion „UpdateObject“ erfolgreich aufgerufen wurde, wird bei Erhalt der Antwort die Methode „Sink_UpdateObject“ aufgerufen.

Event: OnStateVariable_ContainerUpdateIDs Eingabe: ContentDirectory sender, String NewValue

Ausgabe: void

Dieses Event wird geworfen, falls eine Protokoll-Änderung auf dem durch ein Objekt dieser Klasse gekapselten Device registriert wurde. Die betroffene UPnP-Status-Variable heißt „ContainerUpdateIDs“.

Event: OnStateVariable_SystemUpdateID Eingabe: ContentDirectory sender, UInt32 NewValue

Ausgabe: void

Dieses Event wird geworfen, falls eine Protokoll-Änderung auf dem durch ein Objekt dieser Klasse gekapselten Device registriert wurde. Die betroffene UPnP-Status-Variable heißt „SystemUpdateID“.

Event: OnStateVariable_TransferIDs Eingabe: ContentDirectory sender, String NewValue

Ausgabe: void

Dieses Event wird geworfen, falls eine Protokoll-Änderung auf dem durch ein Objekt dieser Klasse gekapselten Device registriert wurde. Die betroffene UPnP-Status-Variable heißt „TransferIDs“.

Event: OnSubscribe Eingabe: ContentDirectory sender, bool Success

Ausgabe: void

Dieses Event wird geworfen, sobald eine Registrierung am ContentDirectory-Service des entsprechenden Devices durchgeführt wurde. Das Flag „Success“ signalisiert einen Erfolg oder aber einen Misserfolg.

Methode: _subscribe Sichtbarkeit: public

Eingabe: int Timeout

Ausgabe: void

Mit Hilfe dieser Methode erfolgt eine Subscription für den Dienst „ContentDirectory“. Für weitere Details bzgl. des Subscription-Mechanismus wird auf die UPnP- bzw. UPnP-AV-Spezifikation verwiesen.

Methode: _subscribe_sink Sichtbarkeit: public

Eingabe: UPnPService sender, bool OK

Ausgabe: void

Diese Methode wird als Callback bei einer erfolgten Subscription aufgerufen.

Methode: Browse Sichtbarkeit: public

Eingabe: String ObjectID, Enum_A_ARG_TYPE_BrowseFlag BrowseFlag, String Filter, UInt32 StartingIndex, UInt32 RequestedCount, String SortCriteria

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „Browse“.

Methode: Browse Sichtbarkeit: public

Eingabe: String ObjectID, Enum_A_ARG_TYPE_BrowseFlag BrowseFlag, String Filter, UInt32 StartingIndex, UInt32 RequestedCount, String SortCriteria, object _Tag, Delegate_OnResult_Browse_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „Browse“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist.

Methode: ContainerUpdateIDs.ModifiedSink Sichtbarkeit: protected

Eingabe: UPnPStateVariable Var, object NewValue

Ausgabe: void

Diese Methode fungiert als Callback, wenn durch Aufruf einer UPnP-Aktion die ContainerUpdateID modifiziert wurde und damit ein neuer Wert für „ContainerUpdateID“ vorhanden ist.

Methode: ContentDirectory Sichtbarkeit: public

Eingabe: UPnPService _S

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor der Klasse „ContentDirectory“. Übergeben wird eine UPnPService-Instanz, die mit diesem Objekt verknüpft wird.

Methode: CreateObject Sichtbarkeit: public

Eingabe: String ContainerID, String Elements

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „CreateObject“.

Methode: CreateObject Sichtbarkeit: public

Eingabe: String ContainerID, String Elements, object _Tag, Delegate_OnResult_CreateObject_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „CreateObject“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist.

Methode: CreateReference Sichtbarkeit: public

Eingabe: String ContainerID, String ObjectID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „CreateReference“.

Methode: CreateReference Sichtbarkeit: public

Eingabe: String ContainerID, String ObjectID, object _Tag, Delegate_OnResult_CreateReference_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „CreateReference“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist.

Methode: DeleteResource Sichtbarkeit: public

Eingabe: Uri ResourceURI

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „DeleteResource“.

Methode: DeleteResource Sichtbarkeit: public

Eingabe: Uri ResourceURI, object _Tag, Delegate_OnResult_DeleteResource_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „DeleteResource“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist.

Methode: DestroyObject Sichtbarkeit: public

Eingabe: String ObjectID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „DestroyObject“.

Methode: DestroyObject Sichtbarkeit: public

Eingabe: String ObjectID, object _Tag, Delegate_OnResult_DestroyObject _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „DestroyObject“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist.

Methode: Dispose Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Mit Hilfe dieser Methode werden „Aufräum“-Operationen durchgeführt. Es erfolgt eine Abmeldung von den zahlreichen Events, sowie eine offizielle UPnP-konforme Un-Subscription-Mitteilung an das betroffene Gerät, um zu signalisieren, dass zukünftige Änderungen bzgl. des Geräte-Zustandes nicht mehr an diesen Kontrollpunkt propagiert werden sollen.

Methode: Enum_A_ARG_TYPE_BrowseFlag_ToString Sichtbarkeit: public

Eingabe: Enum_A_ARG_TYPE_BrowseFlag en

Ausgabe: string

Diese Methode gibt, ausgehend von dem als Parameter übergebenen BrowseFlag, ein Enum_A_ARG_TYPE_BrowseFlag als Zeichenkette zurück. Die Ausgabe kann entweder „BrowseMetadata“ oder „BrowseDirectChildren“ oder ein anderes „unspezifiziertes“ BrowseFlag enthalten.

Methode: Enum_A_ARG_TYPE_TransferStatus_ToString Sichtbarkeit: public

Eingabe: Enum_A_ARG_TYPE_TransferStatus en

Ausgabe: string

Diese Methode gibt, ausgehend von dem als Parameter übergebenen TransferStatus, ein Enum_A_ARG_TYPE_TransferStatus als Zeichenkette zurück. Die Ausgabe kann entweder „COMPLETED“, „ERROR“, „IN_PROGRESS“, „STOPPED“ oder ein anderes „unspezifizierter“ TransferStatus enthalten.

Methode: Error_Sink_Browse Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „Browse“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_CreateObject Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag)

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „CreateObject“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_CreateReference Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „CreateReference“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_DeleteResource Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „DeleteResource“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_DestroyObject Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „DestroyObject“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_ExportResource Sichtbarkeit:private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „ExportResource“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetSearchCapabilities Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetSearchCapabilities“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetSortCapabilities Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetSortCapabilities“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetSystemUpdateID Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetSystemUpdateID“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetTransferProgress Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetTransferProgress“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_ImportResource Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „ImportResource“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_Search Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „Search“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_StopTransferResource Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „StopTransferResource“ zu einem Fehler kommen,

so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_UpdateObject Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „UpdateObject“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: ExportResource Sichtbarkeit: public

Eingabe: Uri SourceURI, Uri DestinationURI

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „ExportResource“.

Methode: ExportResource Sichtbarkeit: public

Eingabe: Uri SourceURI, Uri DestinationURI, object _Tag, Delegate_OnResult_ExportResource_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „ExportResource“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist.

Methode: GetSearchCapabilities Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „GetSearchCapabilities“.

Methode: GetSearchCapabilities Sichtbarkeit: public

Eingabe: object _Tag, Delegate_OnResult_GetSearchCapabilities_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetSearchCapabilities“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist.

Methode: GetSortCapabilities Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „GetSortCapabilities“.

Methode: GetSortCapabilities Sichtbarkeit: public

Eingabe: object _Tag, Delegate_OnResult_GetSortCapabilities_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetSortCapabilities“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist.

Methode: GetSystemUpdateID Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „GetSystemUpdateID“.

Methode: GetSystemUpdateID Sichtbarkeit: public

Eingabe: object _Tag, Delegate_OnResult_GetSystemUpdateID_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetSystemUpdateID“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist.

Methode: GetTransferProgress Sichtbarkeit: public

Eingabe: UInt32 TransferID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „GetTransferProgress“.

Methode: GetTransferProgress Sichtbarkeit: public

Eingabe: UInt32 TransferID, object _Tag, Delegate_OnResult_GetTransferProgress_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetTransferProgress“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist.

Methode: GetUnspecifiedValue Sichtbarkeit: public

Eingabe: string EnumType

Ausgabe: string

Diese Methode gibt den unter dem als Parameter übergebenen Schlüssel gespeicherten Wert in der aktuellen Thread-Hashtabelle zurück.

Methode: GetUPnPService Sichtbarkeit: public

Eingabe: void

Ausgabe: UPnPService

Mit Hilfe dieser Methode besteht die Möglichkeit, auf die durch ein Objekt dieser Klasse gekapselte UPnPService-Instanz zuzugreifen.

Methode: ImportResource Sichtbarkeit: public

Eingabe: Uri SourceURI, Uri DestinationURI

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „ImportResource“.

Methode: ImportResource Sichtbarkeit: public

Eingabe: Uri SourceURI, Uri DestinationURI, object _Tag, Delegate_OnResult_ImportResource_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „ImportResource“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist.

Methode: Search Sichtbarkeit: public

Eingabe: String ContainerID, String SearchCriteria, String Filter, UInt32 StartingIndex, UInt32 RequestedCount, String SortCriteria

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „Search“.

Methode: Search Sichtbarkeit: public

Eingabe: String ContainerID, String SearchCriteria, String Filter, UInt32 StartingIndex, UInt32 RequestedCount, String SortCriteria, object _Tag, Delegate_OnResult_Search_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „Search“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist.

Methode: SetUnspecifiedValue Sichtbarkeit: public

Eingabe: string EnumType, string val

Ausgabe: void

Diese Methode setzt den Wert unter dem als Parameter übergebenen Schlüssel „EnumType“ in der Thread-Hashtabelle auf den neuen Wert „val“.

Methode: Sink_Browse Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „Browse“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_Browse“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_CreateObject Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „CreateObject“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_CreateObject“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_CreateReference Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „CreateReference“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_CreateReference“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_DeleteResource Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „DeleteResource“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_DeleteResource“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_DestroyObject Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „DestroyObject“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_DestroyObject“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_ExportResource Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „ExportResource“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_ExportResource“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetSearchCapabilities Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetSearchCapabilities“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetSearchCapabilities“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetSortCapabilities Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetSortCapabilities“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetSortCapabilities“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetSystemUpdateID Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetSystemUpdateID“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetSystemUpdateID“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetTransferProgress Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetTransferProgress“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird

per Event „Delegate_OnResult_GetTransferProgress“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_ImportResource Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „ImportResource“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_ImportResource“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_Search Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „Search“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_Search“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_StopTransferResource Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „StopTransferResource“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_StopTransferResource“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_UpdateObject Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „UpdateObject“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_UpdateObject“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: StopTransferResource Sichtbarkeit: public

Eingabe: UInt32 TransferID

Ausgabe: void

Durch diese Methode wird die Datenübertragung, die durch die UPnP-Aktionen „ExportResource“ bzw. „ImportResource“ aufgerufen wurde, angehalten. Diese Datenübertragung ist eindeutig mit „TransferID“ identifiziert.

Methode: Sync.Browse Sichtbarkeit: public

Eingabe: String ObjectID, Enum_A_ARG_TYPE_BrowseFlag BrowseFlag, String Filter, UInt32 StartingIndex, UInt32 RequestedCount, String SortCriteria, out String Result, out UInt32 NumberReturned, out UInt32 TotalMatches, out UInt32 UpdateID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „Browse“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variablen „ObjectID“, „BrowseFlag“, „Filter“, „StartingIndex“, „RequestedCount“, „SortCriteria“, „Result“, „NumberReturned“, „TotalMatches“ und „UpdateID“ als Referenzen übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync.CreateObject Sichtbarkeit: public

Eingabe: String ContainerID, String Elements, out String ObjectID, out String Result

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „CreateObject“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variablen „ContainerID“, „Elements“, „ObjectID“ und „Result“ als Referenzen übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync.CreateReference Sichtbarkeit: public

Eingabe: String ContainerID, String ObjectID, out String NewID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „CreateReference“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variablen „ContainerID“, „ObjectID“ und „NewID“ als Referenzen übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync.DeleteResource Sichtbarkeit: public

Eingabe: Uri ResourceURI

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „DeleteResource“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variable „ResourceURI“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_DestroyObject Sichtbarkeit: public

Eingabe: String ObjectID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „DestroyObject“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variable „ObjectID“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_ExportResource Sichtbarkeit: public

Eingabe: Uri SourceURI, Uri DestinationURI, out UInt32 TransferID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „ExportResource“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variablen „SourceURI“, „DestinationURI“ und „TransferID“ als Referenzen übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync_GetSearchCapabilities Sichtbarkeit: public

Eingabe: out String SearchCaps

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetSearchCapabilities“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variable „SearchCaps“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetSortCapabilities Sichtbarkeit: public

Eingabe: out String SortCaps

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetSortCapabilities“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variable „SortCaps“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetSystemUpdateID Sichtbarkeit: public

Eingabe: out UInt32 Id

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetSystemUpdateID“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variable „Id“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetTransferProgress Sichtbarkeit: public

Eingabe: UInt32 TransferID, out Enum_A_ARG_TYPE_TransferStatus TransferStatus, out String TransferLength, out String TransferTotal

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetTransferProgress“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variablen „TransferID“, „TransferStatus“, „TransferLength“ und „TransferTotal“ als Referenzen übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync_ImportResource Sichtbarkeit: public

Eingabe: Uri SourceURI, Uri DestinationURI, out UInt32 TransferID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „ImportResource“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variablen „SourceURI“, „DestinationURI“ und „TransferID“ als Referenzen übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync_Search Sichtbarkeit: public

Eingabe: String ContainerID, String SearchCriteria, String Filter, UInt32 StartingIndex, UInt32 RequestedCount, String SortCriteria, out String Result, out UInt32 NumberReturned, out UInt32 TotalMatches, out UInt32 UpdateID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „Search“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variablen „ContainerID“, „SearchCriteria“, „Filter“, „StartingIndex“, „RequestedCount“, „SortCriteria“, „Result“, „NumberReturned“, „TotalMatches“ und „UpdateID“ als Referenzen übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync_StopTransferResource Sichtbarkeit: public

Eingabe: UInt32 TransferID

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „StopTransferResource“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variable „TransferID“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_UpdateObject Sichtbarkeit: public

Eingabe: String ObjectID, String CurrentTagValue, String NewTagValue

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „UpdateObject“ auf dem Device, das dieser „ContentDirectory“-Instanz zugeordnet ist. Da die Variablen „ObjectID“, „CurrentTagValue“ und „NewTagValue“ als Referenzen übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: SystemUpdateID_ModifiedSink Sichtbarkeit: protected

Eingabe: UPnPStateVariable Var, object NewValue

Ausgabe: void

Wurde eine Protokoll-Änderung an einem Device registriert, so wird die vorliegenden Methode per Event gerufen. Diese Methode wiederum leitet die neuen Informationen per Event „OnStateVariable.SystemUpdateID“ an dort registrierte Methoden weiter.

Methode: TransferIDs_ModifiedSink Sichtbarkeit: protected

Eingabe: UPnPStateVariable Var, object NewValue

Ausgabe: void

Wurde eine Protokoll-Änderung an einem Device registriert, so wird die vorliegenden Methode per Event gerufen. Diese Methode wiederum leitet die neuen Informationen per Event „OnStateVariable.TransferIDs“ an dort registrierte Methoden weiter.

Methode: UpdateObject Sichtbarkeit: public

Eingabe: String ObjectID, String CurrentTagValue, String NewTagValue

Ausgabe: void

Diese Methode wird aufgerufen, wenn ein Metadata_Objekt modifiziert, eingefügt oder gelöscht werden soll. „ObjectID“ identifiziert das Objekt, das modifiziert, eingefügt oder gelöscht wird. „CurrentTagValue“ bzw. „NewTagValue“ sind die zu dem Objekt gehörende jetzige bzw. neue XML_Fragmente.

Property: A_ARG_TYPE_BrowseFlag Sichtbarkeit: public

Eingabe: void

Ausgabe: Enum_A_ARG_TYPE_BrowseFlag

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A_ARG_TYPE_BrowseFlag“ zurück.

Property: A_ARG_TYPE_Count Sichtbarkeit: public

Eingabe: void

Ausgabe: UInt32

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A_ARG_TYPE_Count“ zurück.

Property: A_ARG_TYPE_Filter Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A_ARG_TYPE_Filter“ zurück.

Property: A_ARG_TYPE_Index Sichtbarkeit: public

Eingabe: void

Ausgabe: UInt32

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A.-ARG_TYPE_Index“ zurück.

Property: A_ARG_TYPE_ObjectID Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A.-ARG_TYPE_ObjectID“ zurück.

Property: A_ARG_TYPE_Result Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A.-ARG_TYPE_Result“ zurück.

Property: A_ARG_TYPE_SearchCriteria Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A.-ARG_TYPE_SearchCriteria“ zurück.

Property: A_ARG_TYPE_SortCriteria Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A.-ARG_TYPE_SortCriteria“ zurück.

Property: A_ARG_TYPE_TagValueList Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A.-ARG_TYPE_TagValueList“ zurück.

Property: A_ARG_TYPE_TransferID Sichtbarkeit: public

Eingabe: void

Ausgabe: UInt32

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A.-ARG_TYPE_TransferID“ zurück.

Property: A_ARG_TYPE_TransferLength Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A-ARG_TYPE_TransferLength“ zurück.

Property: A_ARG_TYPE_TransferStatus Sichtbarkeit: public

Eingabe: void

Ausgabe: Enum_A_ARG_TYPE_TransferStatus

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A-ARG_TYPE_TransferStatus“ zurück.

Property: A_ARG_TYPE_TransferTotal Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A-ARG_TYPE_TransferTotal“ zurück.

Property: A_ARG_TYPE_UpdateID Sichtbarkeit: public

Eingabe: void

Ausgabe: UInt32

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A-ARG_TYPE_UpdateID“ zurück.

Property: A_ARG_TYPE_URI Sichtbarkeit: public

Eingabe: void

Ausgabe: Uri

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „A-ARG_TYPE_URI“ zurück.

Property: ContainerUpdateIDs Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „ContainerUpdateIDs“ zurück.

Property: HasAction_Browse Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „Browse“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasAction_CreateObject Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „CreateObject“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasAction_CreateReference Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „CreateReference“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasAction_DeleteResource Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „DeleteResource“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasAction_DestroyObject Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „DestroyObject“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasAction_ExportResource Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „ExportResource“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasAction_GetSearchCapabilities Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetSearchCapabilities“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasAction_GetSortCapabilities Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Aktion „GetSortCapabilities“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasAction_GetSystemUpdateID Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetSystemUpdateID“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasAction_GetTransferProgress Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetTransferProgress“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasAction_ImportResource Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „ImportResource“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasAction_Search Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „Search“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasAction_StopTransferResource Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „StopTransferResource“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasAction_UpdateObject Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Aktion „UpdateObject“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_BrowseFlag Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_BrowseFlag“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_Count Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_Count“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_Filter Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_Filter“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_Index Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_Index“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_ObjectID Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_ObjectID“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_Result Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_Result“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_SearchCriteria Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_SearchCriteria“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_SortCriteria Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_SortCriteria“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_TagValueList Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_TagValueList“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_TransferID Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_TransferID“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_TransferLength Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_TransferLength“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_TransferStatus Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE.TransferStatus“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_TransferTotal Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE.TransferTotal“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_UpdateID Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE.UpdateID“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_A_ARG_TYPE_URI Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE.URI“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_ContainerUpdateIDs Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „ContainerUpdateIDs“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_SearchCapabilities Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „SearchCapabilities“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_SortCapabilities Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Status-Variable „SortCapabilities“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_SystemUpdateID Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „SystemUpdateID“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: HasStateVariable_TransferIDs Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „TransferIDs“ implementiert, die Teil des UPnP-Service „ContentDirectory“ ist.

Property: SearchCapabilities Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „SearchCapabilities“ zurück.

Property: SortCapabilities Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „SortCapabilities“ zurück.

Property: SystemUpdateID Sichtbarkeit: public

Eingabe: void

Ausgabe: UInt32

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „SystemUpdateID“ zurück.

Property: TransferIDs Sichtbarkeit: public

Eingabe: void

Ausgabe: String

Diese Property liefert die Belegung der intern gespeicherten UPnP-Status-Variablen „TransferIDs“ zurück.

Property: UPnPDevice Sichtbarkeit: public

Eingabe: void

Ausgabe: UPnPDevice

Diese Property liefert das mit diesem Objekt verknüpfte Device in Form einer UPnPDevice-Instanz zurück.

Property: Values_A_ARG_TYPE_BrowseFlag Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Diese Property liefert alle möglichen Werte, die eine Variablen vom Typ A_ARG_TYPE_Browse-Flag annehmen kann.

Property: Values_A_ARG_TYPE_TransferStatus Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Diese Property liefert alle möglichen Werte, die eine Variablen vom Typ A_ARG_TYPE_Transfer-Status annehmen kann.

Property: VERSION Sichtbarkeit: public

Eingabe: void

Ausgabe: UPnPDevice

Diese Property liefert die Versionsnummer des zu verwendenden ContentDirectory-Services zurück.

Klasse: RenderingControl Diese Klasse wird verwendet, um eine vollständige UPnP-AV-konforme Implementierung sicherzustellen. Basierend auf dem durch den Intel-Device-Builder ? erzeugten Quellcode, wurden einige Modifikationen vorgenommen, um multiple Media-Instanzen auf einem Gerät unterstützen zu können, wobei nur eine Instanz dieser Klasse pro Device notwendig ist. Bietet ein Device einen „RenderingControl“-Dienst an, so wird automatisch ein Objekt dieser Klasse erzeugt, mit dessen Hilfe ein Zugriff auf die zahlreichen UPnP-AV-Aktionen dieses Dienstes möglich ist. Ein Objekt dieser Klasse wird verwendet, sobald ein UPnP-AV-Renderer im Netzwerk erscheint, das einen RenderingControl-Service anbietet. Intern verwaltet ein solches Objekt alle Instanz-bezogenen Informationen in einer verschachtelten Hash-Tabelle namens „RenderingControlInstances“. Als Key wird die RenderingControlServiceID verwendet, dessen zugeordneter Wert wiederum einer Hash-Tabelle entspricht. Diese Hash-Tabelle erlaubt einen Zugriff auf die UPnP-Status-Variablen. Als Key wird der Name der Variablen verwendet, wobei der dazugehörige Wert einer „UPnPStateVariable“-Instanz entspricht.

Event: OnResult_GetBlueVideoBlackLevel Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 CurrentBlueVideoBlackLevel, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetBlueVideoBlackLevel“ bzw. „Error_Sink_GetBlueVideoBlackLevel“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetBlueVideoGain Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 CurrentBlueVideoGain, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetBlueVideoGain“ bzw. „Error_Sink_GetBlueVideoGain“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetBrightness Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 CurrentBrightness, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetBrightness“ bzw. „Error_Sink_GetBrightness“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetColorTemperature Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 CurrentColorTemperature, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetColorTemperature“ bzw. „Error_Sink_GetColorTemperature“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetContrast Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 CurrentContrast, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetContrast“ bzw. „Error_Sink_GetContrast“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetGreenVideoBlackLevel Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 CurrentGreenVideoBlackLevel, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetGreenVideoBlackLevel“ bzw. „Error_Sink_GetGreenVideoBlackLevel“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetGreenVideoGain Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 CurrentGreenVideoGain, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetGreenVideoGain“ bzw. „Error_Sink_GetGreenVideoGain“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetHorizontalKeystone Eingabe: RenderingControl sender, UInt32 InstanceID, Int16 CurrentHorizontalKeystone, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetHorizontalKeystone” bzw. „Error_Sink_GetHorizontalKeystone” geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetLoudness Eingabe: RenderingControl sender, UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Boolean CurrentLoudness, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetLoudness” bzw. „Error_Sink_GetLoudness” geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetMute Eingabe: RenderingControl sender, UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Boolean CurrentMute, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetMute” bzw. „Error_Sink_GetMute” geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetRedVideoBlackLevel Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 CurrentRedVideoBlackLevel, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetRedVideoBlackLevel” bzw. „Error_Sink_GetRedVideoBlackLevel” geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetRedVideoGain Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 CurrentRedVideoGain, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetRedVideoGain” bzw. „Error_Sink_GetRedVideoGain” geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetSharpness Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 CurrentSharpness, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetSharpness” bzw. „Error_Sink_GetSharpness” geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetVerticalKeystone Eingabe: RenderingControl sender, UInt32 InstanceID, Int16 CurrentVerticalKeystone, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetVerticalKeystone“ bzw. „Error_Sink_GetVerticalKeystone“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetVolume Eingabe: RenderingControl sender, UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, UInt16 CurrentVolume, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetVolume“ bzw. „Error_Sink_GetVolume“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetVolumeDB Eingabe: RenderingControl sender, UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Int16 CurrentVolume, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetVolumeDB“ bzw. „Error_Sink_GetVolumeDB“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_GetVolumeDBRange Eingabe: RenderingControl sender, UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Int16 MinValue, Int16 MaxValue, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_GetVolumeDBRange“ bzw. „Error_Sink_GetVolumeDBRange“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_ListPresets Eingabe: RenderingControl sender, UInt32 InstanceID, String CurrentPresetNameList, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_ListPresets“ bzw. „Error_Sink_ListPresets“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SelectPreset Eingabe: RenderingControl sender, UInt32 InstanceID, Enum_A_ARG_TYPE_PresetName PresetName, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_SelectPreset“ bzw. „Error_Sink_SelectPreset“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetBlueVideoBlackLevel Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 DesiredBlueVideoBlackLevel, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_SetBlueVideoBlackLevel“ bzw. „Error_Sink_SetBlueVideoBlackLevel“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetBlueVideoGain Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 DesiredBlueVideoGain, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_SetBlueVideoGain“ bzw. „Error_Sink_SetBlueVideoGain“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetBrightness Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 DesiredBrightness, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_SetBrightness“ bzw. „Error_Sink_SetBrightness“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetColorTemperature Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 DesiredColorTemperature, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_SetColorTemperature“ bzw. „Error_Sink_SetColorTemperature“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetContrast Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 DesiredContrast, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_SetContrast“ bzw. „Error_Sink_SetContrast“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetGreenVideoBlackLevel Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 DesiredGreenVideoBlackLevel, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_SetGreenVideoBlackLevel“ bzw. „Error_Sink_SetGreenVideoBlackLevel“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetGreenVideoGain Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 DesiredGreenVideoGain, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_SetGreenVideoGain“ bzw. „Error_Sink_SetGreenVideoGain“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetHorizontalKeystone Eingabe: RenderingControl sender, UInt32 InstanceID, Int16 DesiredHorizontalKeystone, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_SetHorizontalKeystone“ bzw. „Error_Sink_SetHorizontalKeystone“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetLoudness Eingabe: RenderingControl sender, Enum_A_ARG_TYPE_Channel Channel, Boolean DesiredLoudness, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_SetLoudness“ bzw. „Error_Sink_SetLoudness“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetMute Eingabe: RenderingControl sender, UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Boolean DesiredMute, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_SetMute“ bzw. „Error_Sink_SetMute“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetRedVideoBlackLevel Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 DesiredRedVideoBlackLevel, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_SetRedVideoBlackLevel“ bzw. „Error_Sink_SetRedVideoBlackLevel“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetRedVideoGain Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 DesiredRedVideoGain, UPnPInvokeException e, object _Tag
Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink_SetRedVideoGain“ bzw. „Error_Sink_SetRedVideoGain“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetSharpness Eingabe: RenderingControl sender, UInt32 InstanceID, UInt16 DesiredSharpness, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink.SetSharpness“ bzw. „Error_Sink.SetSharpness“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetVerticalKeystone Eingabe: RenderingControl sender, UInt32 InstanceID, Int16 DesiredVerticalKeystone, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink.SetVerticalKeystone“ bzw. „Error_Sink.SetVerticalKeystone“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetVolume Eingabe: RenderingControl sender, UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, UInt16 DesiredVolume, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink.SetVolume“ bzw. „Error_Sink.SetVolume“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnResult_SetVolumeDB Eingabe: RenderingControl sender, UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Int16 DesiredVolume, UPnPInvokeException e, object _Tag

Ausgabe: void

Dieses Event wird beim Aufruf der Methode „Sink.SetVolumeDB“ bzw. „Error_Sink.SetVolumeDB“ geworfen, falls eine Methode sich für dieses Ereignis registriert hat.

Event: OnStateVariable_LastChange Eingabe: RenderingControl sender, CommonStateInfo stateInfo

Ausgabe: void

Dieses Event leitet per „CommonStateInfo“-Instanz die neu eingetroffene LastChange-Notifikation an den AVSessionManager weiter. Gerufen wird die Methode „LastChangesRenderingControl“, die sich in der AVSessionManager-Instanz befindet.

Event: OnSubscribe Eingabe: RenderingControl sender, bool Success

Ausgabe: void

Dieses Event wird geworfen, sobald eine Registrierung am RenderingControl-Service des entsprechenden Devices durchgeführt wurde. Das Flag „Success“ signalisiert einen Erfolg oder aber einen Misserfolg.

Methode: _subscribe Sichtbarkeit: public

Eingabe: int Timeout

Ausgabe: void

Mit Hilfe dieser Methode erfolgt eine Subscription für den Dienst „RenderingControl“.

Für weitere Details bzgl. des Subscription-Mechanismus wird auf die UPnP- bzw. UPnP-AV-Spezifikation verwiesen.

Methode: `_subscribe_sink` Sichtbarkeit: protected

Eingabe: UPnPService sender, bool OK

Ausgabe: void

Diese Methode wird als Callback bei einer erfolgten Subscription aufgerufen und leitet die eingegangene Information mit Hilfe des Events „OnSubscribe“ weiter.

Methode: `Dispose` Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Mit Hilfe dieser Methode können „Aufräum“-Operationen durchgeführt werden. Es erfolgt eine Abmeldung von den zahlreichen Events, sowie eine offizielle UPnP-konforme Un-Subscription-Mitteilung an das betroffene Gerät, um zu signalisieren, dass zukünftige Änderungen des Geräte-Zustandes nicht mehr an diesen Kontrollpunkt propagiert werden sollen.

Methode: `Enum_A_ARG_TYPE_Channel_ToString` Sichtbarkeit: public

Eingabe: Enum_A_ARG_TYPE_Channel en

Ausgabe: string

Diese Methode wandelt einen vorliegenden Enumerations-Wert „en“ in einen entsprechenden String um.

Methode: `Enum_A_ARG_TYPE_PresetName_ToString` Sichtbarkeit: public

Eingabe: Enum_A_ARG_TYPE_PresetName en

Ausgabe: string

Diese Methode wandelt einen vorliegenden Enumerations-Wert „en“ in einen entsprechenden String um.

Methode: `Error_Sink_GetBlueVideoBlackLevel` Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetBlueVideoBlackLevel“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: `Error_Sink_GetBlueVideoGain` Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetBlueVideoGain“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetBrightness Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetBrightness“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetColorTemperature Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetColorTemperature“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetContrast Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetContrast“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetGreenVideoBlackLevel Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetGreenVideoBlackLevel“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetGreenVideoGain Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetGreenVideoGain“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetHorizontalKeystone Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvoke

keException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetHorizontalKeystone“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetLoudness Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetLoudness“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetMute Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetMute“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetRedVideoBlackLevel Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetRedVideoBlackLevel“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetRedVideoGain Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetRedVideoGain“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetSharpness Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetSharpness“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetVerticalKeystone Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetVerticalKeystone“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetVolume Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetVolume“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_GetRedVideoGain Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „GetRedVideoGain“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_ListPresets Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „ListPresets“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SelectPreset Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SelectPreset“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetBlueVideoBlackLevel Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetBlueVideoBlackLevel“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetBlueVideoGain Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetBlueVideoGain“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetBrightness Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetBrightness“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetColorTemperature Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetColorTemperature“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetContrast Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetContrast“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetGreenVideoBlackLevel Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetGreenVideoBlackLevel“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetGreenVideoGain Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetGreenVideoGain“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetHorizontalKeystone Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetHorizontalKeystone“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetLoudness Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetLoudness“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetMute Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetMute“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetRedVideoBlackLevel Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetRedVideoBlackLevel“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetRedVideoGain Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetRedVideoGain“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetSharpness Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetSharpness“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetVerticalKeystone Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetVerticalKeystone“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetVolume Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetVolume“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: Error_Sink_SetVolumeDB Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, UPnPInvokeException e, object _Tag

Ausgabe: void

Sollte es im Falle des Aktions-Aufrufs „SetVolumeDB“ zu einem Fehler kommen, so erfolgt der Aufruf dieser Methode. Per Event wird der aufgetretene Fehler weitergeleitet.

Methode: GetA_ARG_TYPE_Channel Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: Enum_A_ARG_TYPE_Channel

Diese Methode liefert den verwendeten „Channel“ der durch „rscID“ referenzierten Medien-Wiedergabe zurück.

Methode: GetA_ARG_TYPE_InstanceID Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: UInt32

Diese Methode liefert die verwendeten „InstanceID“ der durch „rscID“ referenzierten Medien-Wiedergabe zurück.

Methode: GetA_ARG_TYPE_PresetName Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: Enum_A_ARG_TYPE_PresetName

Diese Methode liefert die verwendeten „PresetName“ der durch „rscID“ referenzierten Medien-Wiedergabe zurück.

Methode: GetBlueVideoBlackLevel Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: UInt16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „BlueVideoBlackLevel“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetBlueVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetBlueVideoBlackLevel“.

Methode: GetBlueVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetBlueVideoBlackLevel_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetBlueVideoBlackLevel“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetBlueVideoGain Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: UInt16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „BlueVideoGain“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetBlueVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetBlueVideoGain“.

Methode: GetBlueVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetBlueVideoGain_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetBlueVideoGain“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetBrightness Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: UInt16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „Brightness“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetBrightness Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetBrightness“.

Methode: GetBrightness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetBrightness _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetBrightness“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetColorTemperature Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: UInt16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „ColorTemperature“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetColorTemperature Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetColorTemperature“.

Methode: GetColorTemperature Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetColorTemperature _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetColorTemperature“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetContrast Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: UInt16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „Contrast“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetContrast Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetContrast“.

Methode: GetContrast Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetContrast _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetContrast“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetGreenVideoBlackLevel Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: UInt16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „GreenVideoBlackLevel“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetGreenVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetGreenVideoBlackLevel“.

Methode: GetGreenVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetGreenVideoBlackLevel _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetGreenVideoBlackLevel“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetGreenVideoGain Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: UInt16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „VideoGain“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetGreenVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetGreenVideoGain“.

Methode: GetGreenVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetGreenVideoGain _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetGreenVideoGain“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetHorizontalKeystone Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: Int16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „HorizontalKeystone“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetHorizontalKeystone Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetHorizontalKeystone“.

Methode: GetHorizontalKeystone Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetHorizontalKeystone_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetHorizontalKeystone“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetLastChange Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „LastChange“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetLoudness Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: Boolean

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „Loudness“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetLoudness Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetLoudness“.

Methode: GetLoudness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetLoudness_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetLoudness“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetMute Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: Boolean

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „Mute“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetMute Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetMute“.

Methode: GetMute Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetMute _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetMute“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetPresetNameList Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: String

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „PresetNameList“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetRedVideoBlackLevel Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: UInt16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „RedVideoBlackLevel“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetRedVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetRedVideoBlackLevel“.

Methode: GetRedVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetRedVideoBlackLevel _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetRedVideoBlackLevel“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetRedVideoGain Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: UInt16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „RedVideoGain“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetRedVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetRedVideoGain“.

Methode: GetRedVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetRedVideoGain_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetRedVideoGain“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetRenderingControlServiceIDs Sichtbarkeit: public

Eingabe: void

Ausgabe: ArrayList

Diese Methode liefert eine Liste aller zur Zeit aktiven Medien-Wiedergaben in Form einer Liste bestehend aus den RenderingControlServiceIDs zurück.

Methode: GetSharpness Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: UInt16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „Sharpness“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetSharpness Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetSharpness“.

Methode: GetSharpness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_GetSharpness_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetSharpness“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetStateVariable Sichtbarkeit: private

Eingabe: Int32 rscID, String VarName

Ausgabe: Object

Diese Methode dient als interne Hilfsmethode und liefert den Wert einer durch „VarName“ referenzierten UPnP-Status-Variablen zurück.

Methode: GetStateVariables Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: ArrayList

Diese Methode liefert alle zu einer RenderingControl-Instanz bekannten UPnP-Status-Variablen als „ArrayList“, bestehend aus „UPnPStateVariable“-Instanzen zurück.

Methode: GetUnspecifiedValue Sichtbarkeit: public

Eingabe: string EnumType

Ausgabe: string

Methode: GetUPnPService Sichtbarkeit: public

Eingabe: void

Ausgabe: UPnPService

Mit Hilfe dieser Methode besteht die Möglichkeit, auf die durch ein Objekt dieser Klasse gekapselte UPnPService-Instanz zuzugreifen.

Methode: GetVerticalKeystone Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: Int16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „VerticalKeystone“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetVerticalKeystone Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetVerticalKeystone“.

Methode: GetVerticalKeystone Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate.OnResult_GetVerticalKeystone_Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetVerticalKeystone“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetVolume Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: UInt16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „Volume“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetVolume Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetVolume“.

Methode: GetVolume Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, object _Tag, Delegate_OnResult_GetVolume _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetVolume“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetVolumeDB Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: Int16

Diese Methode liefert die passende Belegung der UPnP-Status-Variablen „VolumeDB“ zu der durch „rscID“ angegebenen RenderingControlServiceID.

Methode: GetVolumeDB Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetVolumeDB“.

Methode: GetVolumeDB Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, object _Tag, Delegate_OnResult_GetVolumeDB _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetVolumeDB“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetVolumeDBRange Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „GetVolumeDBRange“.

Methode: GetVolumeDBRange Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, object _Tag, Delegate_OnResult_GetVolumeDBRange _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „GetVolumeDBRange“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: GetVolumeMaximum Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: Int32

Diese Methode liefert als Rückgabewert die maximal mögliche Lautstärke auf dem durch ein Objekt dieser Klasse gekapselte Device.

Methode: GetVolumeMinimum Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: Int32

Diese Methode liefert als Rückgabewert die minimal mögliche Lautstärke auf dem durch ein Objekt dieser Klasse gekapselte Device.

Methode: GetVolumeStep Sichtbarkeit: public

Eingabe: Int32 rscID

Ausgabe: Int32

Diese Methode liefert als Rückgabewert die atomare Schrittgröße, die eine Änderung der Lautstärke auf dem durch ein Objekt dieser Klasse gekapselte Device beschreibt.

Methode: LastChange_ModifiedSink Sichtbarkeit: protected

Eingabe: UPnPStateVariable Var, object NewValue

Ausgabe: void

Diese Methode wird aufgerufen, sobald eine Last-Change-Notifikation empfangen wurde. Durch den Aufruf der Methode „ParseLastChange“ wird die durch die Variable „NewValue“ übergebene Last-Change-XML-Nachricht geparkt und die enthaltenen neuen Variablen-Belegungen werden intern gesetzt.

Methode: ListPresets Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „ListPresets“.

Methode: ListPresets Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_ListPresets _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „ListPresets“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: ParseLastChange Sichtbarkeit: private

Eingabe: string lastChangeValue

Ausgabe: void

Innerhalb dieser Methode erfolgt nun der eigentlich Parse-Vorgang mit Hilfe eines DOM-Parsers. An dieser Stelle könnte aus Optimierungsgründen in einer späteren Version auf einen SAX-Parser zurückgegriffen werden. Zunächst wird die betroffene RenderingControl-Instanz ermittelt, indem die RenderingControlServiceID aus der Nachricht extrahiert wird. Nun kann die dieser RenderingControlServiceID-zugeordneten Hash-Tabelle ermittelt werden, die die Variablen dieser Instanz beinhaltet. Durch den Aufruf der Methode „SetStateVariables“ werden die betroffenen Status-Variablen intern aktualisiert.

Methode: RenderingControl Sichtbarkeit: public

Eingabe: UPnPService s

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor der Klasse „RenderingControl“. Übergeben wird eine UPnPService-Instanz, die mit diesem Objekt verknüpft wird.

Methode: SelectPresets Sichtbarkeit: public

Eingabe: UInt32 InstanceID

Ausgabe: void

Durch den Aufruf dieser Methode erfolgt ein passend-parametrisierter Aufruf der Methode „SelectPresets“.

Methode: SelectPresets Sichtbarkeit: public

Eingabe: UInt32 InstanceID, object _Tag, Delegate_OnResult_SelectPresets _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SelectPresets“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetBlueVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredBlueVideoBlackLevel

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetBlueVideoBlackLevel“.

Methode: SetBlueVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredBlueVideoBlackLevel, object _Tag, Delegate_OnResult_SetBlueVideoBlackLevel _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetBlueVideoBlackLevel“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetBlueVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredBlueVideoGain

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetBlueVideoGain“.

Methode: SetBlueVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredBlueVideoGain, object _Tag, Delegate_OnResult_SetRedVideoGain _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetBlueVideoGain“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetBrightness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredBrightness

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetBrightness“.

Methode: SetBrightness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredBrightness, object _Tag, Delegate_OnResult_SetBrightness _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetBrightness“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetColorTemperature Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredColorTemperature

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetColorTemperature“.

Methode: SetColorTemperature Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredColorTemperature, object _Tag, Delegate_OnResult_SetColorTemperature _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetColorTemperature“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetContrast Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredContrast

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetContrast“.

Methode: SetContrast Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredContrast, object _Tag, Delegate.OnResult_SetContrast _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetContrast“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetGreenVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredGreenVideoBlackLevel

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetGreenVideoBlackLevel“.

Methode: SetGreenVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredGreenVideoBlackLevel, object _Tag, Delegate.OnResult_SetGreenVideoBlackLevel _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetGreenVideoBlackLevel“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetGreenVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredGreenVideoGain

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetGreenVideoGain“.

Methode: SetGreenVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredGreenVideoGain, object _Tag, Delegate.OnResult_SetGreenVideoGain _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetGreenVideoGain“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetHorizontalKeystone Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Int16 DesiredHorizontalKeystone

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetHorizontalKeystone“.

Methode: SetHorizontalKeystone Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Int16 DesiredHorizontalKeystone, object _Tag, Delegate_OnResult_SetHorizontalKeystone _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetHorizontalKeystone“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetLoudness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Boolean DesiredLoudness

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetLoudness“.

Methode: SetLoudness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Boolean DesiredLoudness, object _Tag, Delegate_OnResult_SetLoudness _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetLoudness“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetMute Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Boolean DesiredMute

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetMute“.

Methode: SetMute Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Boolean DesiredMute, object _Tag, Delegate_OnResult_SetMute _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetMute“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetRedVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredRedVideoBlackLevel

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetRedVideoBlackLevel“.

Methode: SetRedVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredRedVideoBlackLevel, object _Tag, Delegate_OnResult_SetRedVideoBlackLevel _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetRedVideoBlackLevel“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetRedVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredRedVideoGain

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetRedVideoGain“.

Methode: SetRedVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredRedVideoGain, object _Tag, Delegate_OnResult_SetRedVideoGain _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetRedVideoGain“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetRenderingControlID Sichtbarkeit: public

Eingabe: Int32 rcsID

Ausgabe: void

Mit Hilfe dieser Methode wird eine neue Hash-Tabelle mit den durch die gekapselten UPnP-Service-Instanz betroffenen UPnP-Status-Variablen erzeugt.

Methode: SetSharpness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredSharpness

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetSharpness“.

Methode: SetSharpness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredSharpness, object _Tag, Delegate_OnResult_SetSharpness _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetSharpness“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetStateVariables Sichtbarkeit: private

Eingabe: Hashtable stateVariables, int instanceID, XmlNode node

Ausgabe: void

Diese Methode erhält als Eingabe die von der Änderung betroffene Hash-Tabelle, die entsprechende RenderingControlServiceID und eine XMLNode-Instanz, die die Änderungsinformationen enthält. Wurde eine Variable erfolgreich aktualisiert, wird mit Hilfe des Events „OnStateVariable_LastChange“ ein Ereignis ausgelöst, das die RenderingControl-Instanz und eine CommonStateInfo-Instanz übergeben bekommt. Dieses CommonStateInfo-Objekt stellt ein 3er-Tupel bestehend aus dem Variablen-Namen, dem neuen Wert und der RenderingControlServiceID.

Methode: SetUnspecifiedValue Sichtbarkeit: public

Eingabe: string EnumType, string val

Ausgabe: void

Methode: SetVerticalKeystone Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Int16 DesiredVerticalKeystone

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetVerticalKeystone“.

Methode: SetVerticalKeystone Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Int16 DesiredVerticalKeystone, object _Tag, Delegate_OnResult_SetVerticalKeystone _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetVerticalKeystone“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetVolume Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, UInt16 DesiredVolume

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetVolume“.

Methode: SetVolume Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, UInt16 DesiredVolume, object _Tag, Delegate_OnResult_SetVolume _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetVolume“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: SetVolumeDB Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Int16 DesiredVolume

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein passend parametrisierter Aufruf der Methode „SetVolumeDB“.

Methode: SetVolumeDB Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Int16 DesiredVolume, object _Tag, Delegate_OnResult_SetVolumeDB _Callback

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein asynchroner Aufruf der UPnP-Aktion „SetVolumeDB“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sink_GetBlueVideoBlackLevel Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetBlueVideoBlackLevel“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetBlueVideoBlackLevel“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetBlueVideoGain Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetBlueVideoGain“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetBlueVideoGain“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetBrightness Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetBrightness“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetBrightness“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetColorTemperature Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object Ret-

Val, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetColorTemperature“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetColorTemperature“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetContrast Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetContrast“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetContrast“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetGreenVideoBlackLevel Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetGreenVideoBlackLevel“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetGreenVideoBlackLevel“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetGreenVideoGain Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetGreenVideoGain“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetGreenVideoGain“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetHorizontalKeystone Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, objectRetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetHorizontalKeystone“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetHorizontalKeystone“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetLoudness Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetLoudness“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetLoudness“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetMute Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetMute“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetMute“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetRedVideoBlackLevel Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetRedVideoBlackLevel“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetRedVideoBlackLevel“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetRedVideoGain Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetRedVideoGain“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetRedVideoGain“ diese Antwort weitergeleitet oder aber

sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetSharpness Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetSharpness“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetSharpness“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetVerticalKeystone Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „_GetVerticalKeystone“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetVerticalKeystone“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetVolume Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetVolume“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetVolume“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetVolumeDB Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetVolumeDB“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetVolumeDB“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_GetVolumeDBRange Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „GetVolumeDBRange“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_GetVolumeDBRange“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_ListPresets Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „ListPresets“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_ListPresets“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SelectPreset Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SelectPreset“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SelectPreset“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetBlueVideoBlackLevel Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetBlueVideoBlackLevel“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetBlueVideoBlackLevel“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetBlueVideoGain Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetBlueVideoGain“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetBlueVideoGain“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetBrightness Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetBrightness“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetBrightness“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetColorTemperature Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetColorTemperature“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetColorTemperature“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetContrast Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetContrast“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetContrast“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetGreenVideoBlackLevel Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetGreenVideoBlackLevel“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder

wird per Event „Delegate_OnResult_SetGreenVideoBlackLevel“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetGreenVideoGain Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetGreenVideoGain“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetGreenVideoGain“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetHorizontalKeystone Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetHorizontalKeystone“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetHorizontalKeystone“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetLoudness Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetLoudness“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetLoudness“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetMute Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetMute“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetMute“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetRedVideoBlackLevel Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetRedVideoBlackLevel“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetRedVideoBlackLevel“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetRedVideoGain Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetRedVideoGain“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetRedVideoGain“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetSharpness Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetSharpness“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetSharpness“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetVerticalKeystone Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetVerticalKeystone“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetVerticalKeystone“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetVolume Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetVolume“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetVolume“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sink_SetVolumeDB Sichtbarkeit: private

Eingabe: UPnPService sender, string MethodName, UPnPArgument[] Args, object RetVal, object _Tag

Ausgabe: void

Diese Methode wird als Callback aufgerufen, wenn von der UPnP-Aktion „SetVolumeDB“ nach einer erfolgten Anfrage eine Antwort empfangen wird. Entweder wird per Event „Delegate_OnResult_SetVolumeDB“ diese Antwort weitergeleitet oder aber sie wird an diejenige Methode weitergereicht, die per Methodenzeiger in das Objekt „_Tag“ eingebettet wurde.

Methode: Sync_GetBlueVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out UInt16 CurrentBlueVideoBlackLevel

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetBlueVideoBlackLevel“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentBlueVideoBlackLevel“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetBlueVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out UInt16 CurrentBlueVideoGain

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetBlueVideoGain“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentBlueVideoGain“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetBrightness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out UInt16 CurrentBrightness

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetBrightness“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentBrightness“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetColorTemperature Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out UInt16 CurrentColorTemperature

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetColorTemperature“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentColorTemperature“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetContrast Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out UInt16 CurrentContrast

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetContrast“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentContrast“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetGreenVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out UInt16 CurrentGreenVideoBlackLevel

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetGreenVideoBlackLevel“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentGreenVideoBlackLevel“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetGreenVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out UInt16 CurrentGreenVideoGain

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetGreenVideoGain“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentGreenVideoGain“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetHorizontalKeystone Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out Int16 CurrentHorizontalKeystone

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetHorizontalKeystone“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentHorizontalKeystone“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetLoudness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum.A_ARG_TYPE_Channel Channel, out Boolean CurrentLoudness

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetLoudness“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentLoudness“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetMute Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, out Boolean CurrentMute

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetMute“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentMute“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetRedVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 CurrentRedVideoBlackLevel

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetRedVideoBlackLevel“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentRedVideoBlackLevel“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetRedVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out UInt16 CurrentRedVideoGain

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetRedVideoGain“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentRedVideoGain“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetSharpness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out UInt16 CurrentSharpness

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetSharpness“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentSharpness“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetVerticalKeystone Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out Int16 CurrentVerticalKeystone

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetVerticalKeystone“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da

die Variable „CurrentVerticalKeystone“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetVolume Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, out UInt16 CurrentVolume

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetVolume“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentVolume“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetVolumeDB Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, out Int16 CurrentVolume

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetVolumeDB“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentVolume“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_GetVolumeDBRange Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, out Int16 MinValue, out Int16 MaxValue

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „GetVolumeDBRange“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variablen „MinValue“ und „MaxValue“ als Referenz übergeben werden, enthalten diese nach dem Aktionsaufruf die empfangenen Belegungen.

Methode: Sync_ListPresets Sichtbarkeit: public

Eingabe: UInt32 InstanceID, out String CurrentPresetNameList

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „ListPresets“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist. Da die Variable „CurrentPresetNameList“ als Referenz übergeben werden, enthält diese nach dem Aktionsaufruf die empfangene Belegung.

Methode: Sync_SelectPreset Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_PresetName PresetName

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SelectPreset“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetBlueVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredBlueVideoBlackLevel

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetBlueVideoBlackLevel“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetBlueVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredBlueVideoGain

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetBlueVideoGain“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetBrightness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredBrightness

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetBrightness“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetColorTemperature Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredBrightness

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetColorTemperature“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetBrightness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredColorTemperature

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetBrightness“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetContrast Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredContrast

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetContrast“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetGreenVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredGreenVideoBlackLevel

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetGreenVideoBlackLevel“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetGreenVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredGreenVideoGain

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetGreenVideoGain“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetHorizontalKeystone Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Int16 DesiredHorizontalKeystone

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetHorizontalKeystone“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetLoudness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Boolean DesiredLoudness

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetLoudness“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetMute Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum_A_ARG_TYPE_Channel Channel, Boolean DesiredMute

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetMute“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetRedVideoBlackLevel Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredRedVideoBlackLevel

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetRedVideoBlackLevel“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetRedVideoGain Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredRedVideoGain

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetRedVideoGain“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetSharpness Sichtbarkeit: public

Eingabe: UInt32 InstanceID, UInt16 DesiredSharpness

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetSharpness“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetVerticalKeystone Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Int16 DesiredVerticalKeystone

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetVerticalKeystone“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetVolume Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum.A_ARG_TYPE_Channel Channel, UInt16 DesiredVolume

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetVolume“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Methode: Sync.SetVolumeDB Sichtbarkeit: public

Eingabe: UInt32 InstanceID, Enum.A_ARG_TYPE_Channel Channel, Int16 DesiredVolume

Ausgabe: void

Mit Hilfe dieser Methode erfolgt ein synchroner Aufruf der UPnP-Aktion „SetVolumeDB“ auf dem Device, das dieser „RenderingControl“-Instanz zugeordnet ist.

Property: HasAction_GetBlueVideoBlackLevel Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetBlueVideoBlackLevel“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetBlueVideoGain Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetBlueVideoGain“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetBrightness Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetBrightness“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetColorTemperature Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetColorTemperature“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetContrast Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetContrast“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetGreenVideoBlackLevel Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetGreenVideoBlackLevel“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetGreenVideoGain Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetGreenVideoGain“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetHorizontalKeystone Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetHorizontalKeystone“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetLoudness Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetLoudness“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetMute Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetMute“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetRedVideoBlackLevel Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetRedVideoBlackLevel“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetRedVideoGain Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetRedVideoGain“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetSharpness Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetSharpness“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetVerticalKeystone Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetVerticalKeystone“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetVolume Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetVolume“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetVolumeDB Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Aktion „GetVolumeDB“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_GetVolumeDBRange Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „GetVolumeDBRange“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_ListPresets Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „ListPresets“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SelectPreset Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SelectPreset“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetBlueVideoBlackLevel Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetBlueVideoBlackLevel“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetBlueVideoGain Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetBlueVideoGain“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetBrightness Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Aktion „SetBrightness“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetColorTemperature Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetColorTemperature“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetContrast Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetContrast“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetGreenVideoBlackLevel Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetGreenVideoBlackLevel“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetGreenVideoGain Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetGreenVideoGain“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetHorizontalKeystone Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetHorizontalKeystone“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetLoudness Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Aktion „SetLoudness“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetMute Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetMute“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetRedVideoBlackLevel Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetRedVideoBlackLevel“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetRedVideoGain Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetRedVideoGain“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetSharpness Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetSharpness“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetVerticalKeystone Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetVerticalKeystone“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetVolume Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Aktion „SetVolume“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasAction_SetVolumeDB Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Aktion „SetVolumeDB“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_A_ARG_TYPE_Channel Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_Channel“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_A_ARG_TYPE_InstanceID Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_InstanceID“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_A_ARG_TYPE_PresetName Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „A_ARG_TYPE_PresetName“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_BlueVideoBlackLevel Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „BlueVideoBlackLevel“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_BlueVideoGain Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Status-Variable „BlueVideoGain“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_Brightness Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „Brightness“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_ColorTemperature Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „ColorTemperature“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_Contrast Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „Contrast“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_GreenVideoBlackLevel Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „GreenVideoBlackLevel“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_GreenVideoGain Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „GreenVideoGain“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_HorizontalKeystone Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Status-Variable „HorizontalKeystone“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_LastChange Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „LastChange“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_Loudness Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „Loudness“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_Mute Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „Mute“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_PresetNameList Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „PresetNameList“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_RedVideoBlackLevel Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „RedVideoBlackLevel“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_RedVideoGain Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser

Klasse gekapselte Device die UPnP-Status-Variable „RedVideoGain“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_Sharpness Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „Sharpness“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_VerticalKeystone Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „VerticalKeystone“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_Volume Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „Volume“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: HasStateVariable_VolumeDB

Sichtbarkeit: public

Eingabe: void

Ausgabe: bool

Durch den Aufruf dieser Property kann ermittelt werden, ob das durch ein Objekt dieser Klasse gekapselte Device die UPnP-Status-Variable „VolumeDB“ implementiert, die Teil des UPnP-Service „RenderingControl“ ist.

Property: UPnPDevice Sichtbarkeit: public

Eingabe: void

Ausgabe: UPnPDevice

Diese Property liefert das mit diesem Objekt verknüpfte Device in Form einer UPnPDevice-Instanz zurück.

Property: Values_A_ARG_TYPE_Channel Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Diese Property liefert alle möglichen Werte, die eine Variablen vom Typ A_ARG_TYPE_Channel annehmen kann.

Property: Values_A_ARG_TYPE_PresetName Sichtbarkeit: public

Eingabe: void

Ausgabe: string[]

Diese Property liefert alle möglichen Werte, die eine Variablen vom Typ A_ARG_TYPE_-PresetName annehmen kann.

Property: VERSION Sichtbarkeit: public

Eingabe: void

Ausgabe: double

Diese Property liefert die Versionsnummer des zu verwendenden Rendering-Service zurück.

Klasse: CDSAlbum Diese Klasse wird verwendet, um ein durch eine DIDL-Lite-Nachricht repräsentiertes CDS-Album in eine Objektstruktur zu zerlegen.

Methode: CDSAlbum Sichtbarkeit: public

Eingabe: XmlNode node

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor dieser Klasse, mit dessen Hilfe aus einem XML-Node-Objekt eine CDSAlbum-Instanz erzeugt wird. Die Implementierung arbeitet mit einer Konstruktorverkettung, die sich aus der Vererbungshierarchie ergibt. Die Klasse „CDSAlbum“ erbt von der Klasse „CDSContainer“. Durch die Übergabe des XML-Node-Objekts an den Konstruktor der Klasse „CDSContainer“, werden alle dort vorhandenen Attribute mit den Werten belegt, die laut UPnP-Spezifikation für ein Element vom Typ CDS-Container verfügbar sein müssen. Die Übergabe des XML-Node-Objekts wird so lange fortgesetzt bis der Konstruktor der Klasse „CDSObject“ erreicht wurde. Wird der eigentliche Konstruktor dieser Klasse ausgeführt, werden die laut UPnP-Spezifikation vorhandenen Attribute verwendet, um die Objekt-Variablen dieser Stufe zu initialisieren.

Property: Contributor Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „contributor“ in einem Objekt der Klasse „CDSAlbum“ gesetzt bzw. ausgelesen werden.

Property: Date Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „date“ in einem Objekt der Klasse „CDSAlbum“ gesetzt bzw. ausgelesen werden.

Property: Description Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „description“ in einem Objekt der Klasse „CDSAlbum“ gesetzt bzw. ausgelesen werden.

Property: LongDescription Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „longDescription“ in einem Objekt der Klasse „CDSAlbum“ gesetzt bzw. ausgelesen werden.

Property: Publisher Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „publisher“ in einem Objekt der Klasse „CDSAlbum“ gesetzt bzw. ausgelesen werden.

Property: StorageMedium Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „storageMedium“ in einem Objekt der Klasse „CDSAlbum“ gesetzt bzw. ausgelesen werden.

Klasse: CDSAudiolItem Diese Klasse wird verwendet, um ein durch eine DIDL-Lite-Nachricht repräsentiertes CDS-AudioItem in eine Objektstruktur zu zerlegen.

Methode: CDSAudiolItem Sichtbarkeit: public

Eingabe: XmlNode node

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor dieser Klasse, mit dessen Hilfe aus einem XML-Node-Objekt eine CDSAudiolItem-Instanz erzeugt wird. Die Implementierung arbeitet mit einer Konstruktorverkettung, die auch aus der Vererbungshierarchie ergibt. Die Klasse „CDSAudiolItem“ erbt von der Klasse „CDSItem“. Durch die Übergabe des XML-Node-Objekts an den Konstruktor der Klasse „CDSItem“, werden alle dort vorhandenen Attribute mit den Werten belegt, die laut der UPnP-Spezifikation für ein Element vom Typ CDS-Item verfügbar sein müssen. Die Übergabe des XML-Node-Objekts wird so lange fortgesetzt bis der Konstruktor der Klasse „CDSObject“ erreicht wurde. Wird der eigentliche Konstruktor dieser Klasse ausgeführt, werden die laut UPnP-Spezifikation vorhandenen Attribute verwendet, um die Objekt-Variablen dieser Stufe zu initialisieren.

Property: Description Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „description“ in einem Objekt der Klasse „CDSAudioItem“ gesetzt bzw. ausgelesen werden.

Property: Genre Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „genre“ in einem Objekt der Klasse „CDSAudioItem“ gesetzt bzw. ausgelesen werden.

Property: Language Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „language“ in einem Objekt der Klasse „CDSAudioItem“ gesetzt bzw. ausgelesen werden.

Property: LongDescription Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „longDescription“ in einem Objekt der Klasse „CDSAudioItem“ gesetzt bzw. ausgelesen werden.

Property: Publisher Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „publisher“ in einem Objekt der Klasse „CDSAudioItem“ gesetzt bzw. ausgelesen werden.

Property: Relation Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „relation“ in einem Objekt der Klasse „CDSAudioItem“ gesetzt bzw. ausgelesen werden.

Property: Rights Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „rights“ in einem Objekt der Klasse „CDSAudioItem“ gesetzt bzw. ausgelesen werden.

Klasse: CDSContainer Diese Klasse wird verwendet, um ein durch eine DIDL-Lite-Nachricht repräsentiertes CDS-Container in eine Objektstruktur zu zerlegen.

Methode: CDSContainer Sichtbarkeit: public

Eingabe: UPnPDevice device, CDSContainer cdsContainer, String id, String title

Ausgabe: void

Methode: CDSContainer Sichtbarkeit: public

Eingabe: XmlNode node

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor dieser Klasse, mit dessen Hilfe aus einem XML-Node-Objekt eine CDSContainer-Instanz erzeugt wird. Die Implementierung arbeitet mit einer Konstruktorverkettung, die auch aus der Vererbungshierarchie ergibt. Die Klasse „CDSContainer“ erbt von der Klasse „CDSObject“. Durch die Übergabe des XML-Node-Objekts an den Konstruktor der Klasse „CDSObject“, werden alle dort vorhandenen Attribute mit den Werten belegt, die laut der UPnP-Spezifikation für ein Element vom Typ CDS-Object verfügbar sein müssen. Die Übergabe des XML-Node-Objekts wird so lange fortgesetzt bis der Konstruktor der Klasse „CDSObject“ erreicht wurde. Wird der eigentliche Konstruktor dieser Klasse ausgeführt, werden die laut UPnP-Spezifikation vorhandenen Attribute verwendet, um die Objekt-Variablen dieser Stufe zu initialisieren.

Property: ChildCount Sichtbarkeit: public

Eingabe: Int32

Ausgabe: Int32

Durch den Aufruf dieser Property kann der Wert der Variablen „childCount“ in einem Objekt der Klasse „CDSContainer“ gesetzt bzw. ausgelesen werden.

Property: CreateClass Sichtbarkeit: public

Eingabe: ArrayList

Ausgabe: ArrayList

Durch den Aufruf dieser Property kann der Wert der Variablen „createClass“ in einem Objekt der Klasse „CDSContainer“ gesetzt bzw. ausgelesen werden.

Property: Searchable Sichtbarkeit: public

Eingabe: bool

Ausgabe: bool

Durch den Aufruf dieser Property kann der Wert der Variablen „searchable“ in einem Objekt der Klasse „CDSContainer“ gesetzt bzw. ausgelesen werden.

Property: SearchClass Sichtbarkeit: public

Eingabe: ArrayList

Ausgabe: ArrayList

Durch den Aufruf dieser Property kann der Wert der Variablen „searchClass“ in einem Objekt der Klasse „CDSContainer“ gesetzt bzw. ausgelesen werden.

Property: UpdateID Sichtbarkeit: public

Eingabe: UInt32

Ausgabe: UInt32

Durch den Aufruf dieser Property kann der Wert der Variablen „updateID“ in einem Objekt der Klasse „CDSContainer“ gesetzt bzw. ausgelesen werden.

Klasse: CDSImageItem Diese Klasse wird verwendet, um ein durch eine DIDL-Lite-Nachricht repräsentiertes CDSImageItem in eine Objektstruktur zu zerlegen.

Methode: CDSImageItem Sichtbarkeit: public

Eingabe: XmlNode node

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor dieser Klasse, mit dessen Hilfe aus einem XML-Node-Objekt eine CDS-ImageItem-Instanz erzeugt wird. Die Implementierung arbeitet mit einer Konstruktorverkettung, die auch aus der Vererbungshierarchie ergibt. Die Klasse „CDSImageItem“ erbt von der Klasse „CDSObject“. Durch die Übergabe des XML-Node-Objekts an den Konstruktor der Klasse „CDSObject“, werden alle dort vorhandenen Attribute mit den Werten belegt, die laut der UPnP-Spezifikation für ein Element vom Typ CDS-Item verfügbar sein müssen. Die Übergabe des XML-Node-Objekts wird so lange fortgesetzt, bis der Konstruktor der Klasse „CDSObject“ erreicht wurde. Wird der eigentliche Konstruktor dieser Klasse ausgeführt, werden die laut UPnP-Spezifikation vorhandenen Attribute verwendet, um die Objekt-Variablen dieser Stufe zu initialisieren.

Property: Description Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „description“ in einem Objekt der Klasse „CDSImageItem“ gesetzt bzw. ausgelesen werden.

Property: Date Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „date“ in einem Objekt der Klasse „CDSImageItem“ gesetzt bzw. ausgelesen werden.

Property: Description Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „description“ in einem Objekt der Klasse „CDSImageItem“ gesetzt bzw. ausgelesen werden.

Property: LongDescription Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „longDescription“ in einem Objekt der Klasse „CDSImageItem“ gesetzt bzw. ausgelesen werden.

Property: Publisher Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „publisher“ in einem Objekt der Klasse „CDSImageItem“ gesetzt bzw. ausgelesen werden.

Property: Rating Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „rating“ in einem Objekt der Klasse „CDSImageItem“ gesetzt bzw. ausgelesen werden.

Property: Rights Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „rights“ in einem Objekt der Klasse „CDSImageItem“ gesetzt bzw. ausgelesen werden.

Property: StorageMedium Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „storageMedium“ in einem Objekt der Klasse „CDSImageItem“ gesetzt bzw. ausgelesen werden.

Klasse: CDSItem Diese Klasse wird verwendet, um ein durch eine DIDL-Lite-Nachricht repräsentiertes CDS-Item in eine Objektstruktur zu zerlegen.

Methode: CDSItem Sichtbarkeit: public

Eingabe: XmlNode node

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor dieser Klasse, mit dessen Hilfe aus einem XML-Node-Objekt eine CDSItem-Instanz erzeugt wird. Die Implementierung arbeitet mit einer Konstruktorverkettung, die auch aus der Vererbungshierarchie ergibt. Die Klasse „CDSItem“ erbt von der Klasse „CDSObject“. Durch die Übergabe des XML-Node-Objekts an den Konstruktor der Klasse „CDSObject“ werden alle dort vorhandenen Attribute mit den Werten belegt, die laut der UPnP-Spezifikation für ein Element vom Typ CDS-Item verfügbar sein müssen. Die Übergabe des XML-Node-Objekts wird

so lange fortgesetzt, bis der Konstruktor der Klasse „CDSObject“ erreicht wurde. Wird der eigentliche Konstruktor dieser Klasse ausgeführt, werden die laut UPnP-Spezifikation vorhandenen Attribute verwendet, um die Objekt-Variablen dieser Stufe zu initialisieren.

Property: RefId Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „refId“ in einem Objekt der Klasse „CDSImageItem“ gesetzt bzw. ausgelesen werden.

Klasse: CDSMovie Diese Klasse wird verwendet, um ein durch eine DIDL-Lite-Nachricht repräsentiertes CDS-Movie in eine Objektstruktur zu zerlegen.

Methode: CDSMovie Sichtbarkeit: public

Eingabe: XmlNode node

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor dieser Klasse, mit dessen Hilfe aus einem XML-Node-Objekt eine CDSMovie-Instanz erzeugt wird. Die Implementierung arbeitet mit einer Konstruktorverkettung, die auch aus der Vererbungshierarchie ergibt. Die Klasse „CDSMovie“ erbt von der Klasse „CDSVideoItem“. Durch die Übergabe des XML-Node-Objekts an den Konstruktor der Klasse „CDSVideoItem“, werden alle dort vorhandenen Attribute mit den Werten belegt, die laut der UPnP-Spezifikation für ein Element vom Typ CDS-VideoItem verfügbar sein müssen. Die Übergabe des XML-Node-Objekts wird so lange fortgesetzt, bis der Konstruktor der Klasse „CDSObject“ erreicht wurde. Wird der eigentliche Konstruktor dieser Klasse ausgeführt, werden die laut UPnP-Spezifikation vorhandenen Attribute verwendet, um die Objekt-Variablen dieser Stufe zu initialisieren.

Property: ChannelName Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „channelName“ in einem Objekt der Klasse „CDSMovie“ gesetzt bzw. ausgelesen werden.

Property: DVDRegionCode Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „dvdRegionCode“ in einem Objekt der Klasse „CDSMovie“ gesetzt bzw. ausgelesen werden.

Property: ScheduledEndTime Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „scheduledEndTime“ in einem Objekt der Klasse „CDSMovie“ gesetzt bzw. ausgelesen werden.

Property: ScheduledStartTime Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „scheduledStartTime“ in einem Objekt der Klasse „CDSMovie“ gesetzt bzw. ausgelesen werden.

Property: StorageMedium Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „storageMedium“ in einem Objekt der Klasse „CDSMovie“ gesetzt bzw. ausgelesen werden.

Klasse: CDSMusicAlbum Diese Klasse wird verwendet, um ein durch eine DIDL-Lite-Nachricht repräsentiertes CDS-MusicAlbum in eine Objektstruktur zu zerlegen.

Methode: CDSMusicAlbum Sichtbarkeit: public

Eingabe: XmlNode node

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor dieser Klasse, mit dessen Hilfe aus einem XML-Node-Objekt eine CDSMusicAlbum-Instanz erzeugt wird. Die Implementierung arbeitet mit einer Konstruktorverkettung, die sich auch aus der Vererbungshierarchie ergibt. Die Klasse „CDSMusicAlbum“ erbt von der Klasse „CDSAlbum“. Durch die Übergabe des XML-Node-Objekts an den Konstruktor der Klasse „CDSAlbum“, werden alle dort vorhandenen Attribute mit den Werten belegt, die laut der UPnP-Spezifikation für ein Element vom Typ CDS-Album verfügbar sein müssen. Die Übergabe des XML-Node-Objekts wird so lange fortgesetzt bis der Konstruktor der Klasse „CDSObject“ erreicht wurde. Wird der eigentliche Konstruktor dieser Klasse ausgeführt, werden die laut UPnP-Spezifikation vorhandenen Attribute verwendet, um die Objekt-Variablen dieser Stufe zu initialisieren.

Property: AlbumArtURI Sichtbarkeit: public

Eingabe: ArrayList

Ausgabe: ArrayList

Durch den Aufruf dieser Property kann der Wert der Variablen „albumArtURI“ in einem Objekt der Klasse „CDSMusicAlbum“ gesetzt bzw. ausgelesen werden.

Property: Artist Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „artist“ in einem Objekt der Klasse „CDSMusicAlbum“ gesetzt bzw. ausgelesen werden.

Property: Genre Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „genre“ in einem Objekt der Klasse „CDSMusicAlbum“ gesetzt bzw. ausgelesen werden.

Property: Producer Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „producer“ in einem Objekt der Klasse „CDSMusicAlbum“ gesetzt bzw. ausgelesen werden.

Property: Toc Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „toc“ in einem Objekt der Klasse „CDSMusicAlbum“ gesetzt bzw. ausgelesen werden.

Klasse: CDSMusicTrack Diese Klasse wird verwendet, um ein durch eine DIDL-Lite-Nachricht repräsentiertes CDS-MusicTrack in eine Objektstruktur zu zerlegen.

Methode: CDSMusicTrack Sichtbarkeit: public

Eingabe: XmlNode node

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor dieser Klasse, mit dessen Hilfe aus einem XML-Node-Objekt eine CDSMusicTrack-Instanz erzeugt wird. Die Implementierung arbeitet mit einer Konstruktorverkettung, die sich auch aus der Vererbungshierarchie ergibt. Die Klasse „CDSMusicTrack“ erbt von der Klasse „CDSAudioItem“. Durch die Übergabe des XML-Node-Objekts an den Konstruktor der Klasse „CDSAudioItem“, werden alle dort vorhandenen Attribute mit den Werten belegt, die laut der UPnP-Spezifikation für ein Element vom Typ CDS-AudioItem verfügbar sein müssen. Die Übergabe des XML-Node-Objekts wird so lange fortgesetzt bis der Konstruktor der Klasse „CDSObject“ erreicht wurde. Wird der eigentliche Konstruktor dieser Klasse ausgeführt, werden die laut UPnP-Spezifikation vorhandenen Attribute verwendet, um die Objekt-Variablen dieser Stufe zu initialisieren.

Property: Album Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „album“ in einem Objekt der Klasse „CDSMusicTrack“ gesetzt bzw. ausgelesen werden.

Property: Artist Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „artist“ in einem Objekt der Klasse „CDSMusicTrack“ gesetzt bzw. ausgelesen werden.

Property: Contributor Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „contributor“ in einem Objekt der Klasse „CDSMusicTrack“ gesetzt bzw. ausgelesen werden.

Property: Date Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „date“ in einem Objekt der Klasse „CDSMusicTrack“ gesetzt bzw. ausgelesen werden.

Property: OriginalTrackNumber Sichtbarkeit: public

Eingabe: Int32

Ausgabe: Int32

Durch den Aufruf dieser Property kann der Wert der Variablen „originalTrackNumber“ in einem Objekt der Klasse „CDSMusicTrack“ gesetzt bzw. ausgelesen werden.

Property: Playlist Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „playlist“ in einem Objekt der Klasse „CDSMusicTrack“ gesetzt bzw. ausgelesen werden.

Property: StorageMedium Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „storageMedium“ in einem Objekt der Klasse „CDSMusicTrack“ gesetzt bzw. ausgelesen werden.

Klasse: CDSObject Diese Klasse wird verwendet, um ein durch eine DIDL-Lite-Nachricht repräsentiertes CDS-Object in eine Objektstruktur zu zerlegen.

Methode: CDSObject Sichtbarkeit: public

Eingabe: void

Ausgabe: void

Methode: CDSObject Sichtbarkeit: public

Eingabe: XmlNode node

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor dieser Klasse, mit dessen Hilfe aus einem XML-Node-Objekt eine CDSObject-Instanz erzeugt wird. Die Implementierung arbeitet mit einer Konstruktorverkettung, die sich auch aus der Vererbungshierarchie ergibt. Die Klasse „CDSObject“ stellt die Oberklasse aller spezialisierten CDS-Elemente dar. Diese Tatsache entspricht der in UPnP-definierten Objektstruktur, die besagt, dass jedes Medium vom Typ CDS-Object ist. Aufgrund dieser Vererbungshierarchie sind in den spezialisierten Medien-Objekte alle Variablen per Vererbung verfügbar, die die UPnP-Spezifikation vorsieht. Wird der Konstruktor dieser Klasse ausgeführt, so werden die laut UPnP-Spezifikation vorhandenen Attribute verwendet, um die Objekt-Variablen zu initialisieren. Aufgrund der Konstruktor-Verkettung wird aus Effizienzgründen, das geparste XML-Knoten-Element aus dem XML-Node-Objekt herausgelöscht, damit ein entsprechender String-Vergleich nur noch mit tatsächlich denjenigen Variablen durchgeführt werden muss, die auch tatsächlich laut UPnP-Spezifikation verfügbar sein müssen. Als Beispiel dient der Medien-Titel. Laut UPnP-Spezifikation muss jedes Medien-Objekt einen Titel besitzen, so dass sich dieses Attribut im „CDS-Object“ befindet. Wurde in der Case-Struktur ein solches Tag („dc:title“) in der vom Server übertragenen DIDL-Nachricht gefunden, so wird die Variable „title“ mit dem entsprechenden Wert belegt. Durch das Löschen dieses Knotens ist ein Vergleich dieses Knotens beim Konstruktor-Aufruf auf der folgenden Vererbungsstufe natürlich nicht mehr notwendig, ja sogar sinnlos, da kein positives Match auf dieser Ebene gelingen kann.

Property: Creator Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „creator“ in einem Objekt der Klasse „CDSObject“ gesetzt bzw. ausgelesen werden.

Property: Device Sichtbarkeit: public

Eingabe: UPnPDevice

Ausgabe: UPnPDevice

Durch den Aufruf dieser Property kann der Wert der Variablen „device“ in einem Objekt der Klasse „CDSObject“ gesetzt bzw. ausgelesen werden. Bei diesem Wert handelt es sich um den Server, auf dem das entsprechende Medium gefunden wurde.

Property: Id Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „id“ in einem Objekt der Klasse „CDSObject“ gesetzt bzw. ausgelesen werden.

Property: ParentCDSContainer Sichtbarkeit: public

Eingabe: CDSContainer

Ausgabe: CDSContainer

Durch den Aufruf dieser Property kann der Wert der Variablen „parentCDSContainer“ in einem Objekt der Klasse „CDSObject“ gesetzt bzw. ausgelesen werden.

Property: ParentID Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „parentID“ in einem Objekt der Klasse „CDSObject“ gesetzt bzw. ausgelesen werden.

Property: Res Sichtbarkeit: public

Eingabe: ArrayList

Ausgabe: ArrayList

Durch den Aufruf dieser Property kann der Wert der Variablen „res“ in einem Objekt der Klasse „CDSObject“ gesetzt bzw. ausgelesen werden.

Property: Restricted Sichtbarkeit: public

Eingabe: bool

Ausgabe: bool

Durch den Aufruf dieser Property kann der Wert der Variablen „restricted“ in einem Objekt der Klasse „CDSObject“ gesetzt bzw. ausgelesen werden.

Property: Title Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „title“ in einem Objekt der Klasse „CDSObject“ gesetzt bzw. ausgelesen werden.

Property: UPnPClass Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „upnpClass“ in einem Objekt der Klasse „CDSObject“ gesetzt bzw. ausgelesen werden.

Property: WriteStatus Sichtbarkeit: public

Eingabe: Enum_WriteStatus

Ausgabe: Enum_WriteStatus

Durch den Aufruf dieser Property kann der Wert der Variablen „writeStatus“ in einem Objekt der Klasse „CDSObject“ gesetzt bzw. ausgelesen werden.

Klasse: CDSObjectComparer Diese Klasse erbt von der Klasse „IComparer“ und erlaubt es, zu einem im Konstruktor übergebenen Sortierkriterium, mit Hilfe der Methode „Compare“, eine Sortierung zweier Medien-Objekte vorzunehmen.

Methode: CDSObjectComparer Sichtbarkeit: public

Eingabe: String criteria

Ausgabe: void

Bei dieser Methode handelt es sich um den Konstruktor der Klasse „CDSObjectComparer“. Als Übergabe-Parameter dient ein UPnP-konformer Komma-separierte String, der das Sortierkriterium enthält, z.B: „+dc:title,-dc:creator“. Dieses Kriterium wird per String-Operation in seine Bestandteile zerlegt. Pro enthaltenem Kriterium wird eine Instanz des Structs „SortCriteriaStruct“ erzeugt, das die Ordnung („+“), den Namensraum („dc“) und den Element-Namen („title“) aufnimmt. Diese Instanzen werden in eine Array-Liste namens „sortCriteria“ eingefügt.

Methode: Compare Sichtbarkeit: public

Eingabe: object x, object y

Ausgabe: int

Mit Hilfe dieser Methode kann ein Vergleich zweier Medien-Objekte durchgeführt werden, indem per Reflection alle bekannten Attribut-Felder ermittelt werden. Ein Vergleich der beiden Belegungen dieses Attribut-Wertes beider Objekte ermöglicht eine Entscheidung. Da es sich bei dem angegebenen Sortierkriterium um ein hierarchisches Kriterium handelt, muss bei Gleichheit der ersten Komponente noch die verbleibenden Komponenten durchlaufen werden. Sollte keines dieser Komponenten zu einer Entscheidung führen, so muss die Gleichheit beider Objekte bzgl. des angegebenen Sortierkriteriums beim Erreichen der letzten Stufe bescheinigt werden.

Klasse: CDSObjectList Ein Objekt dieser Klasse ist von zentraler Bedeutung in der gesamten Medien-Verwaltung. Verwendet wird eine ArrayList namens „sortedlistOfCdsObjects“, die eine sortierte Liste aller zu dieser „CDSObjectList“-Instanz gehörenden ContentSessions zur Verfügung stellt.

Methode: CDSObjectList Sichtbarkeit: public

Eingabe: CDSObject parentCdsObject, String sortCriteria, int guiID, Control control

Ausgabe: void

Bei dieser Methode handelt es sich um den Konstruktor der Klasse „CDSObjectList“.

Dieser erzeugt eine Instanz der vorliegenden Klassen. Als Übergabeparameter dient eine „CDSObjectList“-Instanz, die dem Elternknoten innerhalb einer Navigationsstruktur entspricht. Sollte dieser nicht bekannt sein, so kann „null“ als Wert übergeben werden. Ebenso wird ein UPnP-konformer Komma-separierter String, der das Sortierkriterium enthält, z.B. ‚+dc:title,-dc:creator‘ in Form der Variablen „sortCriteria“ übergeben. Die Variablen „guiID“ und „control“ sind zur Vermeidung des bereits angesprochenen GUI-Ausgabe-Problems ebenfalls zu übergeben. Unter Anwendung des übergebenen Sortierkriteriums wird so eine neue „CDSObjectComparer“-Instanz erzeugt.

Methode: Clone Sichtbarkeit: public

Eingabe: void

Ausgabe: object

Diese Methode erlaubt es, eine Kopie der bestehenden „CDSObjectList“-Instanz zu erstellen. Alle Elemente werden per Referenz-Kopie dupliziert, d. h. es erfolgt keine echte Kopie („deep copy“) der vorhandenen Elemente. Um aber eine vollständig losgelöste Listen-Struktur zu erhalten, wird ein neues Objekt vom Typ „ArrayList“ erzeugt und der Variablen „sortedlistOfCdsObjects“ zugewiesen. Das nun entstandene Objekt wird zurückgeliefert. Verwendet wird die Methode im Filter-Bereich.

Methode: PutInList Sichtbarkeit: public

Eingabe: CDSObject cdsObject

Ausgabe: int

Diese Methode stellt die Hauptfunktionalität im Bereich der Medienverwaltung bereit. Mit Hilfe der „CDSObjectComparer“-Instanz erfolgt unter Anwendung einer binären Suche auf der Liste der bereits vorhandenen Medien-Objekte, eine Index-Bestimmung bzgl. Speicherort des neu einzufügenden Mediums innerhalb der bestehenden Liste. Das Resultat der Operation „BinarySearch“ liefert einen Index, der entweder positiv oder negativ ist. Im positiven Fall wurde das Medium bzgl. der „CDSObjectComparer“-Instanz bereits eingefügt. Im negativen Fall ermöglicht eine Komplementbildung des Resultats der Operation „BinarySearch“ die Index-Bestimmung der Position an der das Medium einzufügen ist. Unter Verwendung eines „insert“ wird das Medium an die korrekte Position innerhalb der Liste „sortedlistOfCdsObjects“ eingefügt. Das Resultat dieser Methode ist nun der Index des neu eingefügten Objekts innerhalb der Medienliste. Im Kontext dieses Methodenaufrufs wird nach dem Aufruf von „PutInList“ ein Event an die GUI namens „OnSingleCDSObjectSorted“ geworfen, das der GUI das Medienobjekt sowie den Index übergibt.

Property: Control Sichtbarkeit: public

Eingabe: Control

Ausgabe: Control

Durch den Aufruf dieser Property kann der Wert der Variablen „control“ in einem Objekt der Klasse „CDSObjectList“ gesetzt bzw. ausgelesen werden.

Property: Guild Sichtbarkeit: public

Eingabe: int

Ausgabe: int

Durch den Aufruf dieser Property kann der Wert der Variablen „guild“ in einem Objekt der Klasse „CDSObjectList“ gesetzt bzw. ausgelesen werden.

Property: ParentCDSObject Sichtbarkeit: public

Eingabe: CDSObject

Ausgabe: CDSObject

Durch den Aufruf dieser Property kann der Wert der Variablen „parentCDSObject“ in einem Objekt der Klasse „CDSObjectList“ gesetzt bzw. ausgelesen werden.

Property: SortedCDSObjects Sichtbarkeit: public

Eingabe: ArrayList

Ausgabe: ArrayList

Durch den Aufruf dieser Property kann der Wert der Variablen „sortedCDSObjects“ in einem Objekt der Klasse „CDSObjectList“ gesetzt bzw. ausgelesen werden.

Klasse: CDSVideoItem Diese Klasse wird verwendet, um ein durch eine DIDL-Lite-Nachricht repräsentiertes CDS-VideoItem in eine Objektstruktur zu zerlegen.

Methode: CDSVideoItem Sichtbarkeit: public

Eingabe: XmlNode node

Ausgabe: void

Bei dieser Methode handelt es sich um einen Konstruktor dieser Klasse, mit dessen Hilfe aus einem XML-Node-Objekt eine CDSVideoItem-Instanz erzeugt wird. Die Implementierung arbeitet mit einer Konstruktorverkettung, die sich auch aus der Vererbungshierarchie ergibt. Die Klasse „CDSVideoItem“ erbt von der Klasse „CDSItem“. Durch die Übergabe des XML-Node-Objekts an den Konstruktor der Klasse „CDSItem“, werden alle dort vorhandenen Attribute mit den Werten belegt, die laut UPnP-Spezifikation für ein Element vom Typ CDS-Item verfügbar sein müssen. Die Übergabe des XML-Node-Objekts wird so lange fortgesetzt bis der Konstruktor der Klasse „CDSObject“ erreicht wurde. Wird der eigentliche Konstruktor dieser Klasse ausgeführt, werden die laut UPnP-Spezifikation vorhandenen Attribute verwendet, um die Objekt-Variablen dieser Stufe zu initialisieren.

Property: Actor Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „actor“ in einem Objekt der Klasse „CDSVideoItem“ gesetzt bzw. ausgelesen werden.

Property: Description Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „description“ in einem Objekt der Klasse „CDSVideoItem“ gesetzt bzw. ausgelesen werden.

Property: Director Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „director“ in einem Objekt der Klasse „CDSVideoItem“ gesetzt bzw. ausgelesen werden.

Property: Genre Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „genre“ in einem Objekt der Klasse „CDSVideoItem“ gesetzt bzw. ausgelesen werden.

Property: Language Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „language“ in einem Objekt der Klasse „CDSVideoItem“ gesetzt bzw. ausgelesen werden.

Property: LongDescription Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „longDescription“ in einem Objekt der Klasse „CDSVideoItem“ gesetzt bzw. ausgelesen werden.

Property: Producer Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „producer“ in einem Objekt der Klasse „CDSVideoItem“ gesetzt bzw. ausgelesen werden.

Property: Publisher Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „publisher“ in einem Objekt der Klasse „CDSVideoItem“ gesetzt bzw. ausgelesen werden.

Property: Rating Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „rating“ in einem Objekt der Klasse „CDSVideoItem“ gesetzt bzw. ausgelesen werden.

Property: Relation Sichtbarkeit: public

Eingabe: String

Ausgabe: String

Durch den Aufruf dieser Property kann der Wert der Variablen „relation“ in einem Objekt der Klasse „CDSVideoItem“ gesetzt bzw. ausgelesen werden.