

Riedel Router Control Software (RRCS)

Interface protocol specification

Document Reference:

Stephan Stahn

1 CONTENT:

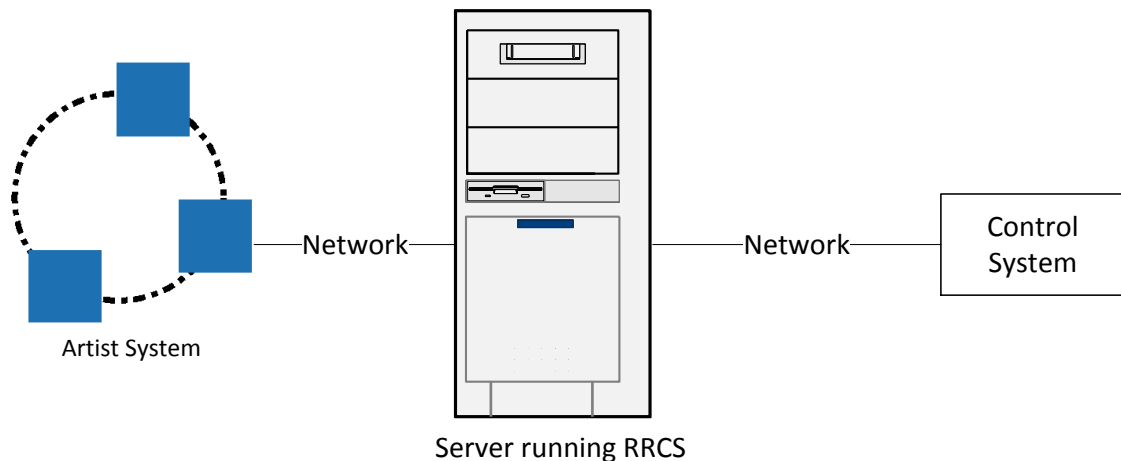
1	CONTENT:.....	2
2	OVERVIEW	5
2.1	Feature list	5
3	DOCUMENT HISTORY.....	6
4	GLOSSARY	12
5	INTERFACE SPECIFICATION	13
5.1	Overview	13
5.2	Physical Layer and data link	13
5.3	Internet Protocol (IP)	13
5.4	Transmission Control Protocol (TCP)	13
5.5	Hypertext Transfer Protocol (HTTP)	13
5.6	Extended Markup Language – Remote Procedure Call (XML-RPC)	14
5.6.1	XML-RPC structure	14
5.6.2	Scalar <value>s	15
5.6.3	<struct>s	15
5.6.4	<array>s	16
5.6.5	XML-RPC Response	16
6	HELPER TYPES.....	21
6.1	TPortAddress	21
6.2	TGroupPortAddress	21
6.3	TConferencePortAddress	23
6.4	TMemberChangeList	24
6.5	Miscellaneous	26
7	ERROR CODES.....	28
8	INCOMING REMOTE PROCEDURES REQUESTS →RRCS	29
8.1	Crosspoints	29
8.2	Volume	31
8.3	Port Alias	32
8.4	Port Label	33
8.5	Input / Output gain	34
8.6	GPIOs	35
8.7	Logic Sources	36
8.8	Status	38
8.9	Objects lists	46
8.9.1	Generic	46
8.9.2	Method GetCommandList	48
8.10	Configuration changes	51
8.10.1	Method ConfigurationChange	51
8.10.2	Method BufferConfigurationChange	54
8.10.3	Method ApplyConfigurationChange	55
8.10.4	Specific configuration parameters	56

8.10.4.1	TCmdPosition	56
8.10.4.2	Object type 'any'	61
8.10.4.3	Object type 'conference'	63
8.10.4.4	Object type 'group'	66
8.10.4.5	Object type 'ifb'	74
8.10.4.6	Object type 'port'	80
8.10.4.7	Object type 'command-container'	88
8.10.4.8	Object type 'call-to-conference'	93
8.10.4.9	Object type 'call-to-group'	105
8.10.4.10	Object type 'call-to-ifb'	116
8.10.4.11	Object type 'dim-level-command'	121
8.10.4.12	Object type 'call-to-port-cmd'	137
8.10.4.13	Object type 'listen-to-port-cmd'	151
8.10.4.14	Object type "route-cmd"	153
8.10.4.15	Object type "switch-gpio-cmd"	155
8.10.4.16	Object type "select-audiopatch-cmd"	156
8.10.4.17	Object type "control-audiopatch-cmd"	157
8.10.4.18	Object type "remote-key-cmd"	158
8.10.4.19	Object type "reply-cmd"	160
8.10.4.20	Object type "edit-conference-cmd"	160
8.10.4.21	Object type "edit-ifb-cmd"	162
8.10.4.22	Object type "dim-panel-speaker-cmd"	163
8.10.4.23	Object type "beep-panel-cmd"	164
8.10.4.24	Object type "telephone-dial-keypad-cmd"	165
8.10.4.25	Object type "telephone-dial-hangup-cmd"	167
8.10.4.26	Object type "logic-cmd"	168
8.10.4.27	Object type "kill-partyline-mic-cmd"	169
8.10.4.28	Object type "auto-listen-off-cmd"	170
8.10.4.29	Object type "set-io-gain-cmd"	171
8.10.4.30	Object type "sidetone-cmd"	172
8.10.4.31	Object type "send-string-cmd"	173
8.10.4.32	Object type "clone-output-port-cmd"	174
8.10.4.33	Object type "panel-key"	175
8.10.4.34	Object type "client-card"	177
8.11	Key and marker manipulation	179
8.12	Panel-Spy	184
8.13	Port Cloning	186
8.14	IFB Volume	188
8.15	Registration for notifications	191
8.15.1	RegisterForAllEvents	196
8.15.2	UnregisterForAllEvents	198
8.16	Trunking	200
8.17	Dial / HangUp of Call	206
9	NOTIFICATIONS	207
9.1	Send String	208
9.2	GPIOs	208
9.3	Logic Source	209
9.4	Volume change	209
9.5	Configuration change	210
9.6	Crosspoint change	211
9.7	Alarms	212
9.8	GetAlive	214

9.9	Panel Spy	215
9.9.1	Overview	215
9.9.2	PanelSpyStateChange	216
9.9.3	PanelSpyRotateEvent	218
9.9.4	PanelSpyKeyEvent	219
9.9.5	PanelSpyFuncKeyEvent	221
9.9.6	PanelSpyNumKeyEvent	222
10	REDUNDANCY	225
11	COMMUNICACTION EXAMPLES.....	227
11.1	Example for an incoming request to RRCS	227
11.2	Example for an outgoing request from RRCS	228
12	PERFORMANCE	230
12.1	Querying information	230
12.2	Pure RRCS communication	231
12.3	Changing system states	231
12.4	Changing system configuration	232

2 OVERVIEW

This document describes RRCS (Riedel Router Control Software), the 3rd party gateway to control the Artist intercom system. RRCS is a software running on Windows XP, Vista or 7. On one side it is connected to one node of an Artist ring, on the other side it provides a network socket with a disclosed protocol described in this document.



Feature list

As a rough overview, RRCS provides the following features to external control systems:

- Setting/Removing crosspoints
- Crosspoint volume control and notifications
- Input / Output gain control
- Set alias / label transfer
- GPI Input change notifications
- Switch GPI outputs
- Logic source control and notifications
- System Status / Alarms
- IFB table control
- Panel key configuration
- Send String notifications
- Using multiple Riedel-panels as external keyboards to control other systems (panel-spy-feature)
- Port Cloning

3 DOCUMENT HISTORY

Version 1.0: 27/01/2004 Initial Draft

Version 1.1: 05/03/2004

Added missing <value> tags to "GetAllActiveXps", "GetAllCaps" and "MonitorTakeConf"

Changed some member names in "GetAllActiveXps" and "GetAllCaps"

Added two new requests "UpstreamFailedCleared" and "DownstreamFailedCleared"

Added net information to most of the failure requests

Added "SetXpPrio"

Version 1.2: 16/03/2007

Added new error codes for invalid audio priority and invalid gpio bit number.

A new member "ErrorCode" has been added to the return structure of the functions:

- GetAllCaps
- GetAllActiveXps

Version 1.3: 06/06/2007

Completed error code list

S/GetInputGain changed: range of gain value is the same as for "S/GetOutputGain" (see "Pacific Quay Software Specification, version 1.4 or higher")

Version 1.4: 21/12/2007

Merged BBC Pacific Quay changes into this document

Added "GetVersion"

Added "CrosspointChange" notification.

Added "RegisterForEventsEx".

Added "UnRegisterForEventsEx".

Version 1.5: 15/01/2008

Renamed struct-member-input-parameter of method "RegisterForEventsEx" from "Alarm" to "General".

Version 1.6 08/04/2008

Added missing method "KillXp"

Version 1.7 28/10/2008

Added performance information

Version 1.8 10/11/2008

Improved documentation for notifications "ClientFailed" and "ClientFailedCleared"

Version 1.9 19/12/2008

Added methods for Logic Sources

Version 2.0 18/10/2009

Changed document layout

Added logic sources

Added object lists

Added configuration changes

Added Volume notifications

Added XP change notifications

Added GetAllIFBs (Artist version >= 5.91)

Added GetAllPorts (Artist version >= 5.91)

Added GetCommandList (Artist version >= 5.91)

Added GetErrorCodeList (Artist version >= 5.91)

Added IsConnectedToArtist (Artist version >= 5.91)

Added IsRegisteredForEvents (Artist version >= 5.91)

Version 6.00 26/02/2010

Added documentation for

- 'ApplyConfigurationChange'
- and 'BufferConfigurationChange'

Version 6.10.1 30/06/2010

Added method documentation:

- IsRegisteredForAllEvents
- RegisterForAllEvents
- UnregisterForAllEvents
- ChangePanelSpy

Added notifications:

- GetAlive
- PanelSpyStateChanged
- PanelSpyRotateEvent
- PanelSpyKeyEvent
- PanelSpyFuncKeyEvent
- PanelSpyNumKeyEvent

Added error code 26 (limit exceeded)

Version 6.20.1 30/07/2010

- possibility to create all available artist commands

Version 6.20.2 14/09/2010

- The IFB properties 'Input', 'Output' and 'MixMinus' can be groups (see 8.10.4.5)
- The command position can now also be a virtual key (see 8.10.4.1)
- The new object type 'command-container' allows to delete all commands from a given command position (see 8.10.4.7)

Version 6.20.3 21/09/2010

- The helper type TGroupPortAddress (see 6.2) may contain the member 'IsInput' to address both: matrix outputs and inputs (which is a new feature since artist v6.20)

Version 6.30.1 08/03/2011

- Port Cloning

Version 6.30.3 14/04/2011

- IFB Volume

Version 6.40.5 05/04/2012

- New function GetAllLogicSources_v2

Version 6.50.2 21/04/2012

- RRCS alarms supported according to the new Artist alarm system. Outgoing alarm notifications have not changed.

Version 6.60.2 11/03/2013

- Compatibility with Release 6.60

Version 6.70.1 24/04/2014

- Compatibility with Release 6.70

Version 6.70.2 06/10/2014

- Added "Set alias / label transfer" to the features list

Version 6.80.1 27/04/2016

- Compatibility with Release 6.80
- Added "Set label" to the features list
- Added "Port Label" section

Version 6.90.1 22/04/2016

- Compatibility with 6.90
- Support for VoIP parameters (client card & port)
- Added "GetTrunkPorts" Added "GetTrunklineSetup"
- Added "GetTrunklineActivities"

Version 6.90.3 10/11/2016

- Bugfix: RRCS Specification: TrunkPortAddr and TrunkNetAddr has to be TrunkingPortAddr and TrunkingNetAddr
- Added "GetTrunkIfbs"
- Bugfix: RRCS Specification: XpVolumeChangeAdd must be XpVolumeChangeRegistryAdd
- Pdf document now has bookmarks
- Bugfix: RRCS Specification: change Boolean to boolean due to RRCS is case sensitive and copying from specification may not work directly.

Version 6.90.4 07/06/2017

- Bugfix: RRCS Specification: Client card VoIP properties are not accessible
- Bugfix: RRCS Specification: Port VoIP properties property type column is messed up.

Version 6.90.5 18/09/2017

- Bugfix: Values of Boolean type in XML-RPC example described in section 8.10.4.12 (call-to-port-cmd) are now excellent correctly

Version 7.10.1 07/06/2017

- ConfigurationChange supports editing properties of port type SIP Phone, RSP, DSP, Bolero Beltpack, AES67Input and AES67Output. -> See chapter 8.10.4.6
- ConfigurationChange supports editing AES67-Client Card → See chapter 8.10.4.34
- Bugfix: Wrong data type definitions in ClientCard VoIP properties
- Bugfix: VoIP port properties accidentally added to GetAllIFB instead of GetAllPort
- Bugfix: Wrong data type definition in SetXPVolume (bool vs Boolean)

Version 7.10.2 18/09/2017

- Bugfix: Values of Boolean type in XML-RPC example described in section 8.10.4.12 (call-to-port-cmd) are now excellent correctly

Version 7.20.1 05/03/2018

Support for dial functionality

Added DialNumber

Added HangUpCall

Added LineStatus

4 GLOSSARY

8 char. Name key	8 character name, which is typically shown on a Riedel panel
Conference	A communication with n members. If one member is talking the other n-1 members are listening.
Longname	32 character name
Net	An Artist network containing several Nodes
Node	A single Artist Frame with at least one Node Controller and client cards
Object ID	32 Bit unique Identifier
Port	A 4-Wire/4-Wire-split audio port or a Panel
RRCS	Riedel Router Control Software
RPC	Remote Procedure Call
XML	Extended Mark-up Language

5 INTERFACE SPECIFICATION

5.1 Overview

The following figure illustrates the protocol structure of the interface. This chapter describes the individual layers in detail.

Ethernet
IP
TCP
HTTP
Messages in XML

5.2 Physical Layer and data link

Generally there aren't any limitations concerning physical network type or topology. However Ethernet is the favorite network type since the gateway applications are running on a standard PC.

5.3 Internet Protocol (IP)

RRCS and RBIS make use of IP. There are no further restrictions concerning the IP-addresses.

5.4 Transmission Control Protocol (TCP)

RRCS and RBIS make use of TCP. The TCP port settings are provisionable. RRCS will listen to TCP Port 8193 per default. RBIS will listen to TCP Port 8194 per default.

5.5 Hypertext Transfer Protocol (HTTP)

RRCS and RBIS make use of HTTP. All messages must be HTTP-POST requests. A User-Agent and Host must be specified. The Content-Type must be text/xml. The Content-Length must be specified and must be correct.

5.6 Extended Markup Language – Remote Procedure Call (XML-RPC)

All messages have to be encoded in XML and follow the XML-RPC specification. What is XML-RPC? It's a spec and a set of implementations that allow software running on disparate operating systems, running in different environments to make procedure calls over a network. It's remote procedure calling using HTTP as the transport and XML as the encoding. XML-RPC is designed to be as simple as possible, while allowing complex data structures to be transmitted, processed and returned

This chapter is a brief description of XML-RPC. Further information can be found on <http://www.xmlrpc.com/>

5.6.1 XML-RPC structure

Example:

```
<?xml version="1.0"?>

<methodCall>

  <methodName>examples.getStateName</methodName>

  <params>

    <param>

      <value><i4>41</i4></value>

    </param>

  </params>

</methodCall>
```

The XML code has a single <methodCall> structure. The <methodCall> must contain a <methodName> sub-item, a string, containing the name of the method to be called. The string may only contain identifier characters, upper and lower-case A-Z, the numeric characters, 0-9, underscore, dot, colon and slash.

If the procedure call has parameters, the <methodCall> must contain a <params> sub-item. The <params> sub-item can contain any number of <param>s, each of which has a <value>.

5.6.2 *Scalar <value>s*

<value>s can be scalars, type is indicated by nesting the value inside one of the tags listed in this table:

Tag	Type	Example
<i4> or <int>	four-byte signed integer	-12
<boolean>	0 (false) or 1 (true)	1
<string>	string	hello world
<double>	double-precision signed floating point number	-12.214
<dateTime.iso8601>	date/time	19980717T14:08:55
<base64>	base64-encoded binary	eW91IGNhbid0IHJlYWQgdGhpcyE=

If no type is indicated, the type is string.

5.6.3 *<struct>s*

A value can also be of type <struct>.

A <struct> contains <member>s and each <member> contains a <name> and a <value>.

Here's an example of a two-element <struct>:

```
<struct>

  <member>

    <name>lowerBound</name>

    <value><i4>18</i4></value>

  </member>

  <member>

    <name>upperBound</name>

    <value><i4>139</i4></value>

  </member>

</struct>
```

<struct>s can be recursive, any <value> may contain a <struct> or any other type, including an <array>, described below.

5.6.4 <array>s

A value can also be of type <array>.

An <array> contains a single <data> element, which can contain any number of <value>s.

Here's an example of a four-element array:

```
<array>
  <data>
    <value><i4>12</i4></value>
    <value><string>Egypt</string></value>
    <value><boolean>0</boolean></value>
    <value><i4>-31</i4></value>
  </data>
</array>
```

<array> elements do not have names.

You can mix types as the example above illustrates.

<arrays>s can be recursive, any value may contain an <array> or any other type, including a <struct>, described above.

5.6.5 XML-RPC Response

Here's an example of a response to an XML-RPC request:

HTTP/1.1 200 OK

Connection: close

Content-Length: 158

Content-Type: text/xml

Date: Fri, 17 Jul 1998 19:55:08 GMT

Server: UserLand Frontier/5.1.2-WinNT

```
<?xml version="1.0"?>
```

```
<methodResponse>
```

```
  <params>
```

```
    <param>
```

```
      <value><string>South Dakota</string></value>
```

```
    </param>
```

```
  </params>
```

```
</methodResponse>
```

Response format:

Unless there's a lower-level error, always return 200 OK.

The Content-Type must be text/xml. Content-Length must be present and correct.

The body of the response is a single XML structure, a `<methodResponse>`, which can contain a single `<params>` which contains a single `<param>` which contains a single `<value>`.

The `<methodResponse>` could also contain a `<fault>` which contains a `<value>` which is a `<struct>` containing two elements, one named `<faultCode>`, an `<int>` and one named `<faultString>`, a `<string>`.

A <methodResponse> can not contain both a <fault> and a <params>.

Fault example:

HTTP/1.1 200 OK

Connection: close

Content-Length: 426

Content-Type: text/xml

Date: Fri, 17 Jul 1998 19:55:02 GMT

Server: UserLand Frontier/5.1.2-WinNT

```
<?xml version="1.0"?>
```

```
<methodResponse>
```

```
  <fault>
```

```
    <value>
```

```
      <struct>
```

```
        <member>
```

```
          <name>faultCode</name>
```

```
          <value><int>4</int></value>
```

```
        </member>
```

```
        <member>
```

```
          <name>faultString</name>
```

```
          <value><string>Too many parameters.</string></value>
```

```
        </member>
```

```
      </struct>
```

```
    </value>
```

</fault>

</methodResponse>

6 HELPER TYPES

Helper types are used in the following chapters as placeholders in order to reduce space in this document.

6.1 *TPortAddress*

The port-address type is used to address Artist ports. The format is as follows:

```
<value><struct>

  <member>

    <name>IsInput</name>

    <value><boolean>{0/1}</boolean></value>

  </member>

  <member>

    <name>Node</name>

    <value><i4>{Node}</i4></value>

  </member>

  <member>

    <name>Port</name>

    <value><i4>{Port}</i4></value>

  </member>

</struct></value>
```

If a null-port is required (e.g.: to clear the mix-minus of an IFB), the Node and Port member must be set to 0.

6.2 *TGroupPortAddress*

The group-port-address type represents a member in an Artist-group.

The format is as follows:

```
<value><struct>
```

<!--note: since artist version 6.20 or later group members can be matrix inputs. Thereto it is necessary to be able to separate input from outputs. To be backward compatible, RRCS adds the 'IsInputOnly' member for pure matrix inputs. -->

```
<member>
```

```
<name>IsInputOnly</name>
```

```
<value><boolean>1</boolean></value>
```

```
</member>
```

```
<member>
```

```
<member>
```

```
<name>Node</name>
```

```
<value><i4>{Node}</i4></value>
```

```
</member>
```

```
<member>
```

```
<name>Port</name>
```

```
<value><i4>{Port}</i4></value>
```

```
</member>
```

```
<member>
```

```
<name>SecondChannel</name>
```

```
<value><boolean>{0/1}</boolean></value>
```

```
</member>
```

```
</struct></value>
```

6.3 *TConferencePortAddress*

The conference-port-address type represents a member in an Artist-conference.

The format is as follows:

```
<value><struct>

  <member>

    <name>Node</name>

    <value><i4>{Node}</i4></value>

  </member>

  <member>

    <name>Port</name>

    <value><i4>{Port}</i4></value>

  </member>

  <member>

    <name>UseSecondChannel </name>

    <value><boolean>{0/1}</boolean></value>

  </member>

  <member>

    <!-- talk right -->

    <name>Talk</name>

    <value><boolean>{0/1}</boolean></value>

  </member>

  <member>

    <!-- listen right -->

    <name>Listen</name>
```

```

<value><boolean>{0/1}</boolean></value>

</member>

</struct></value>

```

6.4 *TMemberChangeList*

This type is used to add and/or remove ports to/from a group. The format is as follows:

```

<value><array><data>

<!-- 1st array element -->

<value><struct>

<!-- tells, whether this port should be added or removed from the
      group-->

<member>

  <name>AddToGroup</name>

  <value><boolean>0</boolean></value>

</member>

<!-- the port address which should be added or removed from the
      Group (notice: this is the type TPortAddress -->

<member>

  <name>PortAddress</name>

  <value><struct>

    <member>

      <name>IsInput</name>

      <value><boolean>0</boolean></value>

```



```
</member>
```

```
<member>
```

```
<name>Node</name>
```

```
<value><i4>{Node}</i4></value>
```

```
</member>
```

```
<member>
```

```
<name>Port</name>
```

```
<value><i4>{Port}</i4></value>
```

```
</member>
```

```
</struct></value>
```

```
</member>
```

```
<!-- tells, whether the 1st or the 2nd channel of the port is used -->
```

```
<member>
```

```
<name>UseSecondChannel</name>
```

```
<value><boolean>{0/1}</boolean></value>
```

```
</member>
```

```
</struct></value>
```

```
<!-- additional array elements -->
```

```
</data></array></value>
```

6.5 Miscellaneous

Expressions in curly brackets used in this chapter are placeholders. The following table is a brief description:

Expression	Description
{Net} {Source Net} {Dest Net} {Net of memb#n}	Artist Network Address. Range 1..255. In most cases there is only one network with address 1.
{Node} {Source Node} {Dest Node} {Node of memb#n}	Artist Node address. Range 2..255.
{Port} {Source Port} {Dest Port} {Port of memb#n}	Artist Port address. Range 0..255. Port address = ((Slot no. – 1) * 8) + Port no. on Client Card -1 e.g. Port 5 on Slot 2 has the port address ((2 – 1) * 8 + 5 – 1) = 12
{Slot-of-client-card}	Client Card slot number. For Artist S the range is 1..4, for Artist M it's 1..16
{Number of XPs}	Number of Crosspoints in the message.
{Priority}	Priority of a crosspoint. Range 0..4 0 = below standard 1 = standard 2 = high 3 = paging 4 = emergency
{Gain}	Input Gain: Range = -36,-35,...,0,...,35,36 gain [dB] = Gain / 2.0
{Status}	Status. Active = 1. Inactive = 0.
{Number of Ports}	Number of Ports in the message.
{IP-address}	IP address in syntax "xxx.xxx.xxx.xxx" or "localhost"
{TCP-Port}	TCP Port number.
{GatewayState}	Gateway status. "Working" or "Standby".
{Volume}	<= 0 → mute 1..255 → (volume-230)/2 dB > 255 → +12.5 dB
{Alias}	Alias name for the given port which is displayed on the panel keys. A string with maximum of 8 characters.
{Slot}	if Port = 128 (GPIO card): 0..15 → bay 1..16 16 → bay X 17 → bay Y 20 → bay B If Port = 0,1,...,127 (Panel GPIO)

	slot-value not relevant
{GPIO no.}	GPIO number 0..15 on this port / slot
{ErrorCode}	Error code. The error code table provides detailed information.
{TransKey}	The transaction key is used in all requests. The response to a request must use the same transaction key. The String has a starting charcater followed by 10 digits. The starting character is "R" if RRCS sends a request. An external Control System can choose a different character. There are no limitations. The 10 digits are free of choice. It's up to the sender of the request how to choose the 10 digits.
{EthernetLinkMode}	Range 1..5 1 = AutoNegotiation 2 = 10 Mbit Half 3 = 10 Mbit Full 4 = 100 Mbit Half 5 = 100 Mbit Full
{TrunkCmdType}	1 = Call to Port 2 = Listen to Port 4 = Call to Conference 5 = Call to Group 48 = Call to IFB 201 = IFB Listen to port 202 = Listen to telex IFB Listen Source 203 = Listen to Artist IFB - MixMinus 204 = Listen to Artist IFB-Input 205 = Conference Call to Conference
{LineStatus}	0 – unknown 1 – Line is free 2 – Waiting for Connection 3 – Line is busy

7 ERROR CODES

Error code	Description
0	Success
1	Transaction key invalid
2	Net address invalid
3	Node address invalid
4	Port address invalid
5	Slot no. invalid
6	Input Gain invalid
7	IP-address invalid
8	TCP-Port invalid
9	Label invalid
10	Conference position invalid
11	Operation failed, because Artist-network not connected
12	Operation failed, because route does not exist
13	Operation not possible, because gateway is standby
14	XML-RPC parameters wrong for this request
15	Invalid conference or conference not found
16	Invalid conference member or conference member not found
17	Invalid priority
18	Invalid GPIO number
19	Invalid gain value
20	Timeout
21	No permission
22	Object does not exist
23	No USB-dongle available
24	Port is not online
25	Object property not supported
26	Limit exceeded
99	Generic error

8 INCOMING REMOTE PROCEDURES REQUESTS →RRCS

8.1 Crosspoints

Description	Remote Procedure	Parameters	Return Value on success
Activate XP source, destination. With standard priority	SetXp	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Source Net}</int></value></param></code> <code><param><value><int>{Source Node}</int></value></param></code> <code><param><value><int>{Source Port}</int></value></param></code> <code><param><value><int>{Dest Net}</int></value></param></code> <code><param><value><int>{Dest Node}</int></value></param></code> <code><param><value><int>{Dest Port}</int></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Activate XP source, destination, priority	SetXpPrio	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Source Net}</int></value></param></code> <code><param><value><int>{Source Node}</int></value></param></code> <code><param><value><int>{Source Port}</int></value></param></code> <code><param><value><int>{Dest Net}</int></value></param></code> <code><param><value><int>{Dest Node}</int></value></param></code> <code><param><value><int>{Dest Port}</int></value></param></code> <code><param><value><int>{Priority}</int></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Activate XP source, destination with audio priority. An existing route to the given destination will be removed. The values Net = 0, Node = 0, port = 0 remove all existing routes for the given source or destination.	SetXpDestructive	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Source Net}</int></value></param></code> <code><param><value><int>{Source Node}</int></value></param></code> <code><param><value><int>{Source Port}</int></value></param></code> <code><param><value><int>{Dest Net}</int></value></param></code> <code><param><value><int>{Dest Node}</int></value></param></code> <code><param><value><int>{Dest Port}</int></value></param></code> <code><param><value><int>{Priority}</int></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Kill all crosspoints between the given	KillXp	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Source Net}</int></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code>

Description	Remote Procedure	Parameters	Return Value on success
source,destination Hint: The method deletes only those crosspoints, which have been activated by RRCS.		<pre> <param><value><int>{Source Node}</int></value></param> <param><value><int>{Source Port}</int></value></param> <param><value><int>{Dest Net}</int></value></param> <param><value><int>{Dest Node}</int></value></param> <param><value><int>{Dest Port}</int></value></param> </pre>	<pre> <value><int>{ErrorCode}</int></value> </data></array></value></param> </pre>
Query status of XP source, destination Hint: The method can return true even if KillXp has been called beforehand.	GetXpStatus	<pre> <param><value><string>{TransKey}</string></value></param> <param><value><int>{Source Net}</int></value></param> <param><value><int>{Source Node}</int></value></param> <param><value><int>{Source Port}</int></value></param> <param><value><int>{Dest Net}</int></value></param> <param><value><int>{Dest Node}</int></value></param> <param><value><int>{Dest Port}</int></value></param> </pre>	<pre> <param><value><array><data> <value><string>{TransKey}</string></value> <value><int>{ErrorCode}</int></value> <value><boolean>{Status}</boolean></value> </data></array></value></param> </pre>
Dump active XPs	GetAllActiveXps	<pre> <param><value><string>{TransKey}</string></value></param> </pre>	<pre> <param><value><struct> <member><name>ErrorCode</name> <value><int>{ ErrorCode }</int></value> </member> <member><name>TransKey</name> <value><string>{TransKey}</string></value> </member> <member><name>XP Count</name> <value><int>{Number of XPs}</int></value> </member> <member><name>XP#1</name> <value><array><data> <value><int>{Source Net}</int></value> <value><int>{Source Node}</int></value> <value><int>{Source Port}</int></value> <value><int>{Dest Net}</int></value> <value><int>{Dest Node}</int></value> <value><int>{Dest Port}</int></value> </data></array></value> </pre>

Description	Remote Procedure	Parameters	Return Value on success
			<pre> </member> <member><name>XP#2</name> . . . </struct></value></param> </pre>
<p>Dumps active crosspoints for the given range.</p> <p>A crosspoint is in range, if the source <u>or</u> (changed in specification version 1.4) the destination of the crosspoint is in range</p> <p>(Start Net <= xp-net <= Stop Net Start Node <= xp-node <= Stop Node Start Port <= xp-port <= Stop Port)</p>	GetActiveXpsRange	<pre> <param><value><string>{TransKey}</string></value></param> <param><value><int>{Start Net}</int></value></param> <param><value><int>{Start Node}</int></value></param> <param><value><int>{Start Port}</int></value></param> <param><value><int>{Stop Net}</int></value></param> <param><value><int>{Stop Node}</int></value></param> <param><value><int>{Stop Port}</int></value></param> </pre>	<p>List of crosspoints. Please see "GetAllActiveXps"</p> <p>HINT: since RRCS version 5500, the return parameters contain an additional error code (see also "GetAllCaps")</p>

8.2 Volume

Description	Remote Procedure	Parameters	Return Value on success
Sets the single or/and conference volume.	SetXpVolume	<pre> <param><value><string>{TransKey}</string></value></param> <param><value><int>{Source Net}</int></value></param> <param><value><int>{Source Node}</int></value></param> </pre>	<pre> <param><value><array><data> <value><string>{TransKey}</string></value> <value><int>{ErrorCode}</int></value> </pre>

Description	Remote Procedure	Parameters	Return Value on success
		<code><param><value><int>{Source Port}</int></value></param></code> <code><param><value><int>{Dest Net}</int></value></param></code> <code><param><value><int>{Dest Node}</int></value></param></code> <code><param><value><int>{Dest Port}</int></value></param></code> <code><param><value><boolean>{single}</bool></value></param></code> <code><param><value>< boolean >{conference}</ boolean ></value></param></code> <code><param><value><int>{volume}</int></value></param></code>	<code></data></array></value></param></code>
Queries the single and conference volume.	GetXpVolume	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Source Net}</int></value></param></code> <code><param><value><int>{Source Node}</int></value></param></code> <code><param><value><int>{Source Port}</int></value></param></code> <code><param><value><int>{Dest Net}</int></value></param></code> <code><param><value><int>{Dest Node}</int></value></param></code> <code><param><value><int>{Dest Port}</int></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code> <value><int>{single volume}</int></value></code> <code> <value><int>{conference volume}</int></value></code> <code></data></array></value></param></code>

8.3 Port Alias

Description	Remote Procedure	Parameters	Return Value on success
<p>Set the alias for the given port.</p> <p>Hint: To change the port alias it is also possible to use the method 'ConfigurationChange'</p>	SetPortAlias	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><string>{Alias}</string></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>

Description	Remote Procedure	Parameters	Return Value on success
with the object type 'port' (requires version 5.91 or later)			
Queries the alias for the given port.	GetPortAlias	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code> <value><string>{Alias}</string></value></code> <code></data></array></value></param></code>

8.4 Port Label

Description	Remote Procedure	Parameters	Return Value on success
<p>Set the label for the given port.</p> <p>Hint: To change the port label it is also possible to use the method 'ConfigurationChange' with the object type 'port' (requires version 5.91 or later)</p>	SetPortLabel	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><string>{Label}</string></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>

Description	Remote Procedure	Parameters	Return Value on success
Queries the label for the given port.	GetPortLabel	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code><value><string>{Label}</string></value></code> <code></data></array></value></param></code>

8.5 Input / Output gain

Description	Remote Procedure	Parameters	Return Value on success
Set input gain of port. Value range: -36,-35,...,0,...,35,36 gain [dB] = Gain / 2.0 Gain = -128 → mute	SetInputGain	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><int>{Gain}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Range, see SetInputGain	GetInputGain	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code><value><int>{Gain}</int></value></code> <code></data></array></value></param></code>
Set the output gain for the given port. Range, see SetInputGain	SetOutputGain	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><int>{Gain}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>

Description	Remote Procedure	Parameters	Return Value on success
Queries the output gain for the given port. Range, see SetInputGain	GetOutputGain	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code> <value><int>{Gain}</int></value></code> <code></data></array></value></param></code>
Activates/Deactivates the given GP output. Hint: The method deactivates only GP outputs that have been activated by RRCS.	SetGpOutput	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><int>{Slot}</int></value></param></code> <code><param><value><int>{GPIO no.}</int></value></param></code> <code><param><value><int>{state}</int></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>

8.6 GPIOs

Description	Remote Procedure	Parameters	Return Value on success
Activates/Deactivates the given GP output. Hint: The method deactivates only GP outputs that have been activated by RRCS.	SetGpOutput	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><int>{Slot}</int></value></param></code> <code><param><value><int>{GPIO no.}</int></value></param></code> <code><param><value><int>{state}</int></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Queries the given GP input state	GetGpInputState	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code> <value><bool>{state}</bool></value></code> <code></data></array></value></param></code>

Description	Remote Procedure	Parameters	Return Value on success
		<code><param><value><int>{Slot}</int></value></param></code> <code><param><value><int>{GPIO no.}</int></value></param></code>	<code></data></array></value></param></code>
<p>Queries the given GP output state</p> <p>Hint: The method can return true, even if the GP output has been deactivated by "SetGpOutput "</p>	GetGpOutputState	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><int>{Slot}</int></value></param></code> <code><param><value><int>{GPIO no.}</int></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code> <value><bool>{state}</bool></value></code> <code></data></array></value></param></code>

8.7 Logic Sources

Description	Remote Procedure	Parameters	Return Value on success
<p>Activates/Deactivates the given logic source.</p> <p>Hint: The method deactivates only logic sources that have been activated by RRCS.</p>	SetLogicSource State	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Object ID}</int></value></param></code> <code><param><value><boolean>{state}</boolean></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
<p>Queries all logic sources in the system</p>	GetAllLogicSources	<code><param><value><string>{TransKey}</string></value></param></code>	<code><param><value><struct></code> <code> <member><name>ErrorCode</name></code> <code> <value><int>{ ErrorCode }</int></value></code> <code> </member></code> <code> <member><name>TransKey</name></code>

Description	Remote Procedure	Parameters	Return Value on success
			<pre> <value><string>{TransKey}</string></value> </member> <member><name>LogicSourceCount</name> <value><int>{Number of LS}</int></value> </member> <member><name>LogicSource#1</name> <value><array><data> <value><string>{Longname}</string></value> <value><string>{8 char. label}</string></value> <value><int>{Object ID}</int></value> </data></array></value> </member> <member><name> LogicSource#2</name> . . . </struct></value></param> </pre>
Queries all logic sources in the system including the current state	GetAllLogicSources_v2	<param><value><string>{TransKey}</string></value></param>	<pre> <param><value><struct> <member><name>ErrorCode</name> <value><int>{ ErrorCode }</int></value> </member> <member><name>TransKey</name> <value><string>{TransKey}</string></value> </member> <member><name>LogicSourceCount</name> <value><int>{Number of LS}</int></value> </member> <member><name>LogicSource#1</name> <value><array><data> <value><string>{Longname}</string></value> <value><string>{8 char. label}</string></value> <value><int>{Object ID}</int></value> <value><boolean>{State}</boolean></value> </data></array></value> </member> </pre>

Description	Remote Procedure	Parameters	Return Value on success
			<pre> <member><name> LogicSource#2</name> . . . </struct></value></param> </pre>

8.8 Status

Description	Remote Procedure	Parameters	Return Value on success
Queries all ports in the system	GetAllCaps	<pre> <param><value><string>{TransKey}</string></value></param> </pre>	<pre> <param><value><struct> <member><name>ErrorCode</name> <value><int>{ ErrorCode }</int></value> </member> <member><name>TransKey</name> <value><string>{TransKey}</string></value> </member> <member><name>port count</name> <value><int>{Number of Ports}</int></value> </member> <member><name>port#1</name> <value><array><data> <value><int>{Net}</int></value> <value><int>{Node}</int></value> <value><int>{Port}</int></value> </data></array></value> </member> <member><name>port#2</name> . . </pre>

Description	Remote Procedure	Parameters	Return Value on success
			<pre> </struct></value></param> </pre>
<p>Queries all ports in the system with more detailed information than 'GetAllCaps'. Hint: This version exists since RRCS version 5.91. The returned data can become very big (approximately 1 kByte per port)</p>	GetAllPorts	<pre> <param><value><string>{TransKey}</string></value></param> </pre>	<pre> <param> <value><array><data> <value><string>{TransKey}</string></value> <value><array><data> <!-- 1st port information --> <value><struct> <member> <name>Input</name> <value><boolean>...</boolean></value> </member> <member> <name>KeyCount</name> <value><i4>...</i4></value> </member> <member> <name>Label</name> <value><string>...</string></value> </member> <!-- this member exists only if an alias exists --> <member> <name>Alias</name> <value><string>...</string></value> </member> <member> <name>Name</name> <value><string>...</string></value> </member> <member> <name>Node</name> <value><i4>...</i4></value> </member> </struct> </value> </array></data> </value></array></data> </param> </pre>

Description	Remote Procedure	Parameters	Return Value on success
			<pre> <member> <name>ObjectID</name> <value><i4>...</i4></value> </member> <member> <name>Output</name> <value><boolean>...</boolean></value> </member> <member> <name>PageCount</name> <value><i4>...</i4></value> </member> <member> <name>Port</name> <value><i4>...</i4></value> </member> <member> <name>PortType</name> <value><string>...</string></value> </member> <member> <!--this member exists only if an expansion panel exists --> <name>ConnectedExpansions</name> <value>array of expansion panel object-IDs </value> </member> <member> <name>PortVoIP</name> <value><struct> <member> <name>AudioCodec</name> <value><i4>...</i4></value> </member> </pre>

Description	Remote Procedure	Parameters	Return Value on success
			<pre> <member> <name>AudioPacketSize</name> <value><i4>...</i4></value> </member> <member> <name>DiffServiceCodePoint</name> <value><i4>...</i4></value> </member> <member> <name>LocalSipId</name> <value><string>...</string></value> </member> <member> <name>ReceiveBufferSize</name> <value><i4>...</i4></value> </member> <member> <name>RemoteHost</name> <value><string>{IP-address}</string></value> </member> <member> <name>RemoteSipId</name> <value><string>...</string></value> </member> <member> <name>VoiceActDetection</name> <value><boolean>...</boolean></value> </member> </struct></value> </member> </struct></value> <!--further ports --> ... </data></array></value> </pre>

Description	Remote Procedure	Parameters	Return Value on success
			<pre> </data></array></value> </param> </params> </pre>
<p>Queries all IFB's in Artist.</p> <p>Hint: This version exists since RRCS version 5.91</p>	GetAllIFBs	<pre> <param><value><string>{TransKey}</string></value></param> </pre>	<pre> <param> <value><array><data> <value><string>{TransKey}</string></value> <value><array><data> <!-- 1st IFB information --> <value><struct> <member> <name>Input</name> <value>{TPortAddress}</value> </member> <member> <name>Label</name> <value><string>...</string></value> </member> <member> <name>LongName</name> <value><string>...</string></value> </member> <member> <name>MixMinus</name> <value>{TPortAddress}</value> </member> <member> <name>Number</name> <value><i4>...</i4></value> </member> <member> <name>ObjectID</name> <value><i4>...</i4></value> </member> </member> </array> </data> </value> </param> </pre>

Description	Remote Procedure	Parameters	Return Value on success
			<pre> <name>Output</name> <value>{TPortAddress}</value> </member> <member> <name>Owner</name> <value><i4>...</i4></value> </member> </struct></value> <!-- further IFB information --> ... </data></array></value> </data></array></value> </param> </pre>
Get a list of possible error codes including their error descriptions	GetErrorCodeList	<param><value><string>{TransKey}</string></value></param>	<pre> <param> <value><array><data> <value><string>{TransKey}</string></value> <value><array><data> <value><struct> <member> <name>Code</name> <value><i4>...</i4></value> </member> <member> <name>Description</name> <value><string>...</string></value> </member> </struct></value> <!-- further error information --> ... </data></array></value> </data></array></value> </param> </params> </methodResponse> </pre>

Description	Remote Procedure	Parameters	Return Value on success
Query gateway state	GetState	<param><value><string>{TransKey}</string></value></param>	<param><value><array><data> <value><string>{TransKey}</string></value> <value><int>{ErrorCode}</int></value> <value><string>{GatewayState}</string></value> </data></array></value></param>
Set gateway to Working	SetStateWorking	<param><value><string>{TransKey}</string></value></param>	<param><value><array><data> <value><string>{TransKey}</string></value> <value><int>{ErrorCode}</int></value> </data></array></value></param>
Set gateway to Standby	SetStateStandby	<param><value><string>{TransKey}</string></value></param>	<param><value><array><data> <value><string>{TransKey}</string></value> <value><int>{ErrorCode}</int></value> </data></array></value></param>
Ping the gateway	GetAlive	<param><value><string>{TransKey}</string></value></param>	<param><value><array><data> <value><string>{TransKey}</string></value> <value><int>{ErrorCode}</int></value> </data></array></value></param>
Retreive the RRCS version	GetVersion	<param><value><string>{TransKey}</string></value></param>	<param><value><array><data> <value><string>{TransKey}</string></value> <value><int>{ErrorCode}</int></value> <value><string>{Version}</string></value> </data></array></value></param>
Tests whether the RRCS sends notifications to the given XML-RPC-server or not	IsRegisteredFor Events	<param> <value><string>{TransKey}</string></value> </param> <param> <value><string>{IP-address}</string></value> </param> <param> <value><int>{TCP-port}</int></value>	<param> <value><struct> <member> <name>IsRegistered</name> <value><boolean>...</boolean></value> </member> <member> <name>TransKey</name>

Description	Remote Procedure	Parameters	Return Value on success
		</param>	<value><string>{TransKey}</string></value> </member> </struct></value> </param>
Tests whether RRCS is connected to Artist or not Hint: The method returns 'not connected', if no network connection to Artist is available or the configuration format does not match with the version of RRCS	IsConnectedToArtist	<param><value><string>{TransKey}</string></value></param>	<param> <value><struct> <member> <name>IsConnected</name> <value><boolean>...</boolean></value> </member> <member> <name>TransKey</name> <value><string>{TransKey}</string></value> </member> </struct></value> </param>
Tests whether RRCS sends notifications to the given address or not. Internally the method tests whether RRCS sends notifications to the url ("http://" [ip-address of incoming RPC] ":" [TCPPort] [URLPath]) or not. The url path is optional and is treated as "/RPC2" if left blank	IsRegisteredForAllEvents	<param> <value><string>{TransKey}</string></value> </param> <param> <value><i4>{TCPPort}</i4></value> </param> <param> <value><string>{URLPath}</string></value> </param>	<param> <value><struct> <member> <name>IsRegistered</name> <value><boolean>...</boolean></value> </member> <member> <name>TransKey</name> <value><string>{TransKey}</string></value> </member> </struct></value> </param>

8.9 Objects lists

8.9.1 Generic

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
<p>Gets all existing objects in the system of a specific type</p> <p>Object-type can be:</p> <ul style="list-style-type: none"> - "conference" - "group" - "port" - "ifb" - "logic-source" - "logic-destination" - "gp-input" - "gp-output" - "user" - "audiopatch" - "client-card" <p>The object type is case-insensitive</p>	GetObjectList	<pre> <param> <value> <string>{TransKey}</string> </value> </param> <param> <value> <string>{object-type}</string> </value> </param> </pre>	<pre> <param> <value> <struct> <member> <name>TransKey</name> <value>{TransKey}</value> </member> <member> <name>ObjectList</name> <value> <array> <data> <value> <struct> <member> <name>ObjectID</name> <value>{Object ID}</value> </member> <member> <name>LongName</name> <value>{ LongName}</value> </member> </struct> </value> ... </data> </array> </value> </member> </struct> </value> </param> </pre>

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
			<pre> </member> </struct> </value> </param> </pre>
<p>Gets an object property.</p> <p>Hint: 1) The method "ConfigurationChange" with the object type "any" allows to set any property for any object.</p> <p>2) If the property name is empty, the method returns all available object properties</p>	GetObjectProperty	<pre> <param> <value> <string>{TransKey}</string> </value> </param> <param> <value><int>{Object ID}</int></value> </param> <param> <value> <string>{Property name}</string> </value> </param> </pre>	<pre> <param> <value> <struct> <member> <name>TransKey</name> <value>{TransKey}</value> </member> <member> <name>{Property name}</name> <value> {Property Value} </value> </member> </struct> </value> </param> </pre>
Gets a list of supported object properties.	GetObjectPropertyNames	<pre> <param> <value> <string>{TransKey}</string> </value> </param> <param> <value><int>{Object ID}</int></value> </param> </pre>	<pre> <param> <value> <struct> <member> <name>TransKey</name> <value>{TransKey}</value> </member> <member> <name>PropertyNames</name> <value><array><data> <value>{property name 1}</value> <value>{property name 2}</value> ... </data></array></value> </member> </struct> </value> </param> </pre>

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
			<pre> </struct> </value> </param> </pre>

Object properties can be retrieved by using `GetObjectProperty`. This method does not require an object type since the object ID already provides enough information. However, different object types support different properties. The return type {Property type} depends on the property. E.g. an alias would be a `<string>`

8.9.2 Method *GetCommandList*

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
Queries the command list of a given position. Positon can be a key or virtual function	GetCommandList	<pre> <param> <value><string>{TransKey}</string></value> </param> <param> <{TCmdPosition}> </param> </pre>	<pre> <param> <value><struct> <member> <name>CommandList</name> <value><array><data> <value><struct> <member> <name>Description</name> <value><string>{command description}</string></value> </member> </pre>

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
			<pre> <member> <name>{command Object ID}</name> <value><int>{command object ID}</int></value> </member> </struct></value> <value><struct> <member> <name>Description</name> <value><string>{command description}'</string></value> </member> <member> <name>ObjectID</name> <value><int>{command object ID}</int></value> </member> </struct></value> ... </data></array></value> </member> <member> <name>TransKey</name> <value><string>{TransKey}</string></value> </member> </struct></value> </param> </pre>

Object-type	Supported properties	Property-type
conference	Alias	<string>
	Label	<string> (8 characters max)
	LongName	<string> (32 unicode characters max)
	Owner	<int> (object-ID of user-object)
	MemberList	<array> of <{TConferencePortAddress}> (read-only, see 6.3)
group	Label	<string> (8 characters max)
	LongName	<string> (32 unicode characters max)
	MemberList	<array> of <{TGroupPortAddress}> (see 6.2)
	Owner	<int> (Object-ID of user-object)
ifb	Label	<string> (8 characters max)
	LongName	<string> (32 unicode characters max)
	Number	<int> (read-only)
	Input	<{TPortAddress}> (see 6.1)
	Output	<{TPortAddress}> (see 6.1)
	MixMinus	<{TPortAddress}> (see 6.1)
	Owner	<int> (Object-ID of user-object)

8.10 Configuration changes

All configuration changes (create, replace, edit and delete) will use the method ConfigurationChange. The format of the parameters is flexible enough to support all objects and their properties in the future. For now only “call to conference”, “call to group”, “call to ifb”, “ifb” and “conference” are supported, please see the table. Objects have their specific parameters since the properties are very different. For the specific parameters please see 8.10.2.

8.10.1 Method ConfigurationChange

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
<p>Performs a configuration change.</p> <p>{ ChangeType } can be: “create” “edit” “delete” “delete-children”</p> <p>{ ObjectType } can be: “any” “call-to-conference” “ifb” “conference” “group” “call-to-ifb” “call-to-group” “dim-level-command” “call-to-port-cmd” “listen-to-port-cmd” “route-cmd” “switch-gpio-out-cmd” “select-audiopatch-cmd”</p>	ConfigurationChange	<pre> <param> <value><string>{TransKey}</string></value> </param> <param> <value> <array> <data> <!--1st configuration change --> <value> <struct> <member> <name>ChangeType</name> <value>{ ChangeType }</value> </member> <member> <name>ObjectType</name> <value> <string>{ ObjectType }</string> </value> </member> <member> <name>SpecificParams</name> <value>{ specific parameter struct }</value> </member> </struct> </value> </data> </array> </value> </param> </pre>	<pre> <param> <value> <string>{TransKey}</string> </value> </param> </pre>

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
"control-audiopatch-cmd" "remote-key-cmd" "reply-cmd" "edit-conference-cmd" "edit-ifb-cmd" "dim-panel-speaker-cmd" "beep-panel-cmd" "telephone-dial-keypad-cmd" "logic-cmd" "kill-partyline-mic-cmd" "auto-listen-off-cmd" "set-input-output-gain-cmd" "sidetone-cmd" "send-string-cmd" "command-container"		</member> </struct> </value> <!--2 nd configuration change --> <value> <struct> </struct> </value> <!-- last configuration change --> <value> <struct> </struct> </value> </data> </array></value></param>	

Object-types	ChangeType
call-to-conference	Create, Edit, Delete
conference	Edit
ifb	Edit
group	Edit
call-to-ifb	Create, Edit, Delete
call-to-group	Create, Edit, Delete
dim-level-command	Create, Edit, Delete

call-to-port-cmd	Create, Edit, Delete
listen-to-port-cmd	Create, Edit, Delete
route-cmd	Create, Edit, Delete
switch-gpio-out-cmd	Create, Delete
select-audiopatch-cmd	Create, Delete
control-audiopatch-cmd	Create, Delete
remote-key-cmd	Create, Edit, Delete
reply-cmd	Create, Edit, Delete
edit-conference-cmd	Create, Delete
edit-ifb-cmd	Create, Delete
dim-panel-speaker-cmd	Create, Edit, Delete
beep-panel-cmd	Create, Delete
telephone-dial-keypad-cmd	Create, Edit, Delete
logic-cmd	Create, Delete
kill-partyline-mic-cmd	Create, Delete
auto-listen-off-cmd	Create, Delete
set-input-output-gain-cmd	Create, Delete
sidetone-cmd	Create, Edit, Delete
send-string-cmd	Create, Delete
hot-mic-cmd	Create, Delete
command-container	delete-children
client-card	

8.10.2 Method BufferConfigurationChange

Buffers configuration changes

The caller should call the method 'ApplyConfigurationChange' later to send the buffered configuration changes to Artist. Hint: If the caller does not call 'ApplyConfigurationChange' or 'BufferConfigurationChange' within the next 60 seconds, the server deletes all previously buffer configurations changes. In this case the next call of 'ApplyConfigurationChange' or 'BufferConfigurationChange' will return an error. Once the error has been returned back to the client it is possible to restart buffering.

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
Buffers configuration changes. { ChangeType } can be: "create" "edit" "delete" { ObjectType } can be: "any" "call-to-conference" "ifb" "conference" "group" "call-to-ifb" "call-to-group" "dim-level-command"	BufferConfigurati onChange	<pre> <param> <value><string>{TransKey}</string></value> </param> <param> <value> <array> <data> <!--1st configuration change --> <value> <struct> <member> <name>ChangeType</name> <value>{ ChangeType }</value> </member> <member> <name>ObjectType</name> <value> <string>{ ObjectType }</string> </value> </member> </struct> </value> </data> </array> </value> </param> </pre>	<pre> <param> <value> <string>{TransKey}</string> </value> </param> </pre>

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
		<pre> <name>SpecificParams</name> <value>{ specific parameter struct }</value> </member> </struct> </value> <!--2nd configuration change --> <value> <struct> </struct> </value> <!-- last configuration change --> <value> <struct> </struct> </value> </data> </array></value></param> </pre>	

8.10.3 Method *ApplyConfigurationChange*

Sends buffered configuration changes (via 'BufferConfigurationChange') to Artist

- 'ApplyConfigurationChange' must be called at most 60 seconds after the last 'BufferConfigurationChange' call. Otherwise the server returns a time out error and ignores previously buffered configuration changes.

- If the server buffers nothing the server returns success.

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
Sends buffered configuration changes (via 'BufferConfigurationChange') to Artist	ApplyConfigurationChange	<code><param></code> <code> <value><string>{TransKey}</string></value></code> <code></param></code>	<code><param></code> <code> <value></code> <code> <string>{TransKey}</string></code> <code> </value></code> <code></param></code>

8.10.4 Specific configuration parameters

8.10.4.1 TCmdPosition

TCmdPosition defines the place of a command. Commands can be placed onto a key or a virtual Function of a port. If you want to specify a key, you need to use the members ExpansionPanel, Page and KeyNumber. The member VirtualFunction is nonexistent in this case. If you want to specify a virtual Function, you need to use the member virtual function. In this case ExpansionPanel, Page and KeyNumber are nonexistent.

The member 'PositionType' always exists and can currently have one of the following values:

'key'

'virtual-function'

-'virtual-key'

The number of available position types can grow in the future, e.g. “virtual-key” or “gp-output”

RRCS ignores struct-members that are unnecessary.

Command position is on a key

The member expansion panel only exist, if the is on an expansion panel.

```
<struct>
```

```
<member>
```

```
  <name>PositionType</name>
```

```
  <value><string>key</string></value>
```

```
</member>
```

```
<member>
```

```
  <name>Node</name>
```

```
  <value><int>{ ... }</int></value>
```

```
</member>
```

```
<member>
```

```
  <!-- counting starts with 0 -->
```

```
  <name>Port</name>
```

```
<value><int>{...}</int></value>

</member>

<member>

    <!-- optional, default is 1 -->

    <name>PoolPort</name>

    <value><int>{...}</int></value>

</member>

<member>

    <!-- counting starts with 1 -->

    <name>ExpansionPanel</name>

    <value><int>{...}</int></value>

</member>

<member>

    <!-- counting starts with 1 -->

    <name>Page</name>

    <value><int>{...}</int></value>
```

```
</member>
```

```
<member>
```

```
    <!-- counting starts with 1-->
```

```
    <name>KeyNumber</name>
```

```
    <value><int>{...}</int></value>
```

```
</member>
```

```
<member>
```

```
    <!-- optional, default is false -->
```

```
    <name>IsInput</name>
```

```
    <value><boolean>{...}</boolean></value>
```

```
</member>
```

```
</struct>
```

Command position is on a virtual function

The member 'VirtualFunctionType' can have one of the following values:

'vox'

'always'

'on-call'

<struct>

<member>

<name>PositionType</name>

<value><string>virtual-function</string></value>

</member>

<member>

<name>Node</name>

<value><int>{ ... }</int></value>

</member>

<member>

<name>Port</name>

```
<value><int>{...}</int></value>

</member>

<member>

<name>VirtualFunctionType</name>

    <value><string>{...}</string></value> <!-- e.g. {vox} -->

</member>

<member>

    <!-- optional, default is false -->

    <name>IsInput</name>

    <value><boolean>{...}</boolean></value>

</member>

</struct>
```

8.10.4.2 Object type ‘any’

The object-type “any” can be used to edit properties of any object. It implements the inverse operation of the method “GetObjectProperty”. This object-type is here just for completeness. It is not needed for the Canal+ project.

8.10.4.2.1 *Specific parameters for edit*

```
<member>
  <name>ObjectID</name>
  <value><int>{ ObjectID }</int ></value>
</member>
<member>
  <name>Properties</name>
  <value><struct>
    <!-- 1st property to edit -->
    <member>
      <name>{Property name}</name>
      <value><{Property type}>{Property Value}</{Property type}</value>
    </member>
    ...
  </struct></value>
</member>
```

8.10.4.3 Object type 'conference'

8.10.4.3.1 Specific parameters for editing "conference"

The edit conference functionality requires an object-ID of the conference to modify. The properties of the conference are shown in the table below:

Supported properties	Property-type
Alias	<string>
Label	<string> (8 characters max)
LongName	<string> (32 unicode characters max)
Owner	<int> (object-ID of user-object)

8.10.4.3.2 Example

The following example changes the alias, the label and the long-name of a conference with the object-ID 39153498:

```
<?xml version="1.0"?">
```

```
<methodCall>
```

```
<methodName>ConfigurationChange</methodName>
```

```
<params>
```

```
<param>
```

<value>C00000000005</value>

</param>

<param>

<value><array><data>

<value><struct>

<member>

<name>ChangeType</name>

<value>edit</value>

</member>

<member>

<name>ObjectType</name>

<value>conference</value>

</member>

<member>

<name>SpecificParams</name>

<value><struct>

<member>

<name>ObjectID</name>

<value><i4>39153498</i4></value>

</member>

<member>

<name>Alias</name>

<value>confAlia</value>

</member>

<member>

<name>Label</name>

<value>confLabl</value>

</member>

<member>

<name>LongName</name>

```

        <value>New long-name</value>

      </member>

    </struct></value>

  </member>

</struct></value>

</data></array></value>

</param>

</params>

</methodCall>

```

8.10.4.4 Object type ‘group’

8.10.4.4.1 Specific parameters for editing “group”

The edit group functionality requires an object-ID of the group to modify. The change properties of the group-object are shown in the table below:

Supported properties	Property-type
----------------------	---------------

Label	<string> (8 characters max)
LongName	<string> (32 unicode characters max)
MemberList	<{TMemberChangeList}> (an array of ports to add or remove, see 6.4) Notice: This is not (!) the same type as used by the method "GetObjectProperty"
Owner	<int> (object-ID of user-object)

8.10.4.4.2 *Example*

This example changes the alias, the label, and the long-name of the group with object-ID 17251. Additionally it adds the port with the address (node=2 and port=0) and removes the port with the address (node=2 and port=1).

```
<?xml version="1.0"?>
```

```
<methodCall>
```

```
<methodName>ConfigurationChange</methodName>
```

```
<params>
```

```
<param>
```

```
<value>C00000000005</value>
```

```
</param>
```

<param>

<value><array><data>

<value><struct>

<member>

<name>ChangeType</name>

<value>edit</value>

</member>

<member>

<name>ObjectType</name>

<value>group</value>

</member>

<member>

<name>SpecificParams</name>

<value><struct>

<member>

<name>ObjectID</name>

<value><i4>17251</i4></value>

</member>

<member>

<name>Alias</name>

<value>grpAlias</value>

</member>

<member>

<name>Label</name>

<value>grpLabel</value>

</member>

<member>

<name>LongName</name>

<value>New group long-name </value>

</member>

```
<member>

  <name>MemberList</name>

  <value><array><data>

    <value><struct>

      <member>

        <name>AddToGroup</name>

        <value><boolean>1</boolean></value>

      </member>

    <member>

      <name>PortAddress</name>

      <value><struct>

        <member>

          <name>IsInput</name>

          <value><boolean>0</boolean></value>

        </member>

      <member>
```

```
<name>Node</name>

<value><i4>2</i4></value>

</member>

<member>

  <name>Port</name>

  <value><i4>0</i4></value>

</member>

</struct></value>

</member>

<member>

  <name>UseSecondChannel</name>

  <value><boolean>0</boolean></value>

</member>

</struct></value>

<value><struct>

  <member>
```

```
<name>AddToGroup</name>

<value><boolean>0</boolean></value>

</member>

<member>

  <name>PortAddress</name>

  <value><struct>

    <member>

      <name>IsInput</name>

      <value><boolean>0</boolean></value>

    </member>

    <member>

      <name>Node</name>

      <value><i4>2</i4></value>

    </member>

    <member>

      <name>Port</name>
```



```
<value><i4>1</i4></value>

</member>

</struct></value>

</member>

<member>

  <name>UseSecondChannel</name>

  <value><boolean>0</boolean></value>

</member>

</struct></value>

</data></array></value>

</member>

</struct></value>

</member>

</struct></value>

</data></array></value>

</param>
```

</params>

</methodCall>

8.10.4.5 Object type 'ifb'

8.10.4.5.1 Specific parameters for edit "ifb"

Supported properties	Property-type
LongName	<string> (32 unicode characters max)
Label	<string> (8 characters max)
Input	<{TPortAddress}> or <TIFBAudioEndpoint>
Output	<{TPortAddress}> or <TIFBAudioEndpoint>
MixMinus	<{TPortAddress}> or <TIFBAudioEndpoint>
Owner	<int> (object-ID of user-object)

Hint: If the 'Node' and the 'Port' members of a {TPortAddress} are set to 0, the properties 'Input', 'Output' and 'MixMinus' are set to nothing.

Since RRCS 6.20 the members 'Input', 'Output' and 'MixMinus' can also be artist-groups. Hence these members can now contain the new data type **TIFBAudioEndpoint** which is defined as follows:

TIFBAudioEndpoint

Member	Type						
AudioEndpointType	<string> Can contain one of the following values: <table border="1"> <thead> <tr> <th>Value</th><th>Meaning</th></tr> </thead> <tbody> <tr> <td>Nil</td><td>the audio endpoint is empty</td></tr> <tr> <td>Group</td><td>the endpoint defines an artist group (see member "ObjectID")</td></tr> </tbody> </table>	Value	Meaning	Nil	the audio endpoint is empty	Group	the endpoint defines an artist group (see member "ObjectID")
Value	Meaning						
Nil	the audio endpoint is empty						
Group	the endpoint defines an artist group (see member "ObjectID")						
ObjectID	<int> (AudioEndpointType must be „Group“ (case-insensitive))						

If an audio-endpoint is empty, the method GetObjectProperty returns a TPortAddress with the node and port member set to zero.

This is done for backward compatibility reasons.

8.10.4.5.2 *Example*

Example for editing the label to “Sport” on the IFB number 7

```
<?xml version="1.0"?>
```

```
<methodCall>
```

```
<methodName>ConfigurationChange</methodName>
```

```
<params>
```

```
<param>
```

```
<value>C00000000001</value>
```

```
</param>
```

```
<param>
```

```
<value><array><data>
```

```
<value><struct>
```

```
<member>
```

```
<name>ChangeType</name>
```

```
<value>edit</value>
```

```
</member>
```

```
<member>
```

```
  <name>ObjectType</name>
```

```
  <value>ifb</value>
```

```
</member>
```

```
<member>
```

```
  <name>SpecificParams</name>
```

```
  <value><struct>
```

```
    <!-- Hint: This member addresses the IFB only, it does not change the number -->
```

```
  <member>
```

```
    <name>IFBNumber</name>
```

```
    <value><i4>7</i4></value>
```

```
  </member>
```

```
<member>
```

```
  <name>Label</name>
```

```
<value>Sport</value>
</member>
</struct></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>
```

8.10.4.6 Object type 'port'

8.10.4.6.1 Specific parameters for edit

Parameters	Type
PortAddress	<{TPortAddress}>
Alias	<string>
PoolPort	<int> (optional)
PortVoIP	<struct>
PortSip	<struct>
PortAes67	<struct>
PortAes67Input	<struct>
PortAes67Output	<struct>
PortAes67Bolero	<struct>

PortVoIP (all members are optional)	Property-type
RemoteHost	<string> (optional)
RemoteSipId	<string> (optional)
LocalSipId	<string> (optional)
AudioCodec	<int> (optional) 0 = G.711 U-law 8k 8 = G.711 A-law 8k 9 = G.722 64kbps PLC 84 = G.711 U-law 16k 91 = G.711 A-law 16k 97 = PCM 8k

	110 = RARe U-law 111 = RARe A-law 112 = G.722 48kbps PLC
AudioPacketSize	<int> (optional) 0 = 20 ms 1 = 40 ms 2 = 80 ms 3 = 160 ms
ReceiveBufferSize	<int> (optional) Depending on the AudioPacketSize, Max is 5120 ms 0 = 4 x AudioPacketSize 1 = 8 x AudioPacketSize 2 = 16 x AudioPacketSize 3 = 32 x AudioPacketSize 4 = 64 x AudioPacketSize 5 = 128 x AudioPacketSize 6 = 256 x AudioPacketSize
DiffServiceCodePoint	<int> (optional)
VoiceActDetection	<boolean> (optional)

PortSip (all members are optional)	Property-type
DomainServer	<string> (optional)
ProxyServer	<string> (optional)
UserName	<string> (optional)
DisplayName	<string> (optional)
AuthUserName	<string> (optional)

AuthPassword	<string> (optional)
Reregister	<int> (optional)
SipProtocol	<int> (optional) 0 = TCP 1 = UDP
TrustedDomain	<boolean> (optional)

PortAes67 (all members are optional)	Property-type
IpAddress	{IpAddress} (optional)
ListenPort	<int> (optional)
PacketTime	<int> (optional) 125 = 0.125 ms 250 = 0.250 ms 333 = 0.333 ms 1000 = 1.000 ms 1333 = 1.333 ms
ReceiveBuffer	<int> (optional) N x PacketTime <= 18000 ; where n <= 18

PortAes67Input (all members are optional)	Property-type
BitDepth	<int>(optional) 16 = L16 24 = L24
PacketTime	<int> (optional)

	125 = 0.125 ms 250 = 0.250 ms 333 = 0.333 ms 1000 = 1.000 ms 1333 = 1.333 ms
ReceiveBuffer	<int> (optional) N x PacketTime <= 18000 ; where n <= 18
PayloadType	<int> (optional) 96 - 127
SSRC	<string> (optional)
TimeStampOffset	<string> (optional)
Channels	<int> (optional) 1 2 4 8
Selection	<int> (optional) Depending on the configured channels 1 ..8
Enable2nd	<boolean> (optional)
Protocol	<int> (optional) 2 = Manual 3 = RTSP
SourceIp	{IpAddress} (optional)
Multicast	{IpAddress} (optional)
MulticastPort	<int> (optional)
RTSPUri	<string> (optional)

PortAes67Output (all members are optional)	Property-type
BitDepth	<int>(optional) 16 = L16 24 = L24
PacketTime	<int> (optional) 125 = 0.125 ms 250 = 0.250 ms 333 = 0.333 ms 1000 = 1.000 ms 1333 = 1.333 ms
PayloadType	<int> (optional) 96 - 127
SSRC	<string> (optional)
TimeStampOffset	<string> (optional)
Channels	<int> (optional) 1 2 4 8
Selection	<int> (optional) Depending on the configured channels 1 ..8
Enable2nd	<boolean> (optional)
Protocol	<int> (optional) 2 = Manual 3 = RTSP
Multicast	{IpAddress} (optional)
MulticastPort	<int> (optional)

PortAes67Bolero (all members are optional)	Property-type
BoleroUserId	<int>(optional)
Multicast	{IpAddress} (optional)
MulticastPort	<int> (optional)

Example to change the alias on physical port 0, pool port 1, node 2:

```
<?xml version="1.0"?>
<methodCall>
<methodName>ConfigurationChange</methodName>
<params>
<param>
  <value>C00000000005</value>
</param>
<param>
  <value><array><data>
    <value><struct>
      <member>
        <name>ChangeType</name>
        <value>edit</value>
      </member>
      <member>
        <name>ObjectType</name>
        <value>port</value>
      </member>
      <member>

```

```
<name>SpecificParams</name>
<value><struct>

  <member>
    <name>PortAddress</name>
    <value><struct>
      <member>
        <name>IsInput</name>
        <value><boolean>1</boolean></value>
      </member>
      <member>
        <name>Node</name>
        <value><i4>2</i4></value>
      </member>
      <member>
        <name>Port</name>
        <value><i4>0</i4></value>
      </member>
    </struct></value>
  </member>

  <member>
    <name>PoolPort</name>
    <value><int>1</int></value>
  </member>

  <member>
    <name>Alias</name>
    <value>NewAlias</value>
  </member>
```

```
</struct></value>  
</member>  
</struct></value>  
</data></array></value>  
</param>  
</params>  
</methodCall>
```

8.10.4.7 Object type 'command-container'

This object type allows to delete all commands from a given command position. There's an optional parameter 'RestrictToCreator' where the caller can select the commands which have to be deleted on a creator basis. At present the only filter is RRCS (RestrictToCreator=1) or no filter (RestrictToCreator=0). If the parameter is missing, RRCS will delete all commands.

Example:

```
<?xml version="1.0"?>
<methodCall>
<methodName>ConfigurationChange</methodName>
<params>
<param>
  <value><string>C00000000003</string></value>
</param>
<param>
  <value><array><data>
    <value><struct>
      <member>
        <name>ChangeType</name>
        <value><string>delete-children</string></value>
      </member>
      <member>
        <name>ObjectType</name>
        <value><string>command-container</string></value>
      </member>
      <member>
```



```
<name>SpecificParams</name>
<value><struct>
  <member>
    <name>RestrictToCreator</name>
    <value><int>1</int></value>
  </member>
  <member>
    <name>CommandPosition</name>
    <value><struct>
      <member>
        <name>KeyNumber</name>
        <value><i4>1</i4></value>
      </member>
      <member>
        <name>Node</name>
        <value><i4>2</i4></value>
      </member>
      <member>
        <name>Page</name>
        <value><i4>1</i4></value>
      </member>
      <member>
        <name>Port</name>
        <value><i4>0</i4></value>
      </member>
      <member>
        <name>PositionType</name>
        <value><string>key</string></value>
      </member>
    </struct></value>
  </member>
</struct></value>
```

```

        </struct></value>
      </member>
    </struct></value>
  </data></array></value>
</param>
</params>
</methodCall>

```

This object type can also be used to delete a batch of commands. In this case you can pass a list of object IDs that should be deleted by means of the member “CommandsToDelete”.

Example:

```

<methodCall>
<methodName>
  ConfigurationChange
</methodName>
<params>
<param>
  <value><string>C00000000003</string></value>
</param>
<param>
  <value><array><data>
    <value><struct>
      <member>
        <name>ChangeType</name>
        <value><string>delete-children</string></value>
      </member>

```

```
<member>
  <name>ObjectType</name>
  <value><string>command-container</string></value>
</member>
<member>
  <name>SpecificParams</name>
  <value><struct>
    <member>
      <name>RestrictToCreator</name>
      <value><int>1</int></value>
    </member>
    <member>
      <name>CommandPosition</name>
      <value><struct>
        <member>
          <name>KeyNumber</name>
          <value><i4>1</i4></value>
        </member>
        <member>
          <name>Node</name>
          <value><i4>2</i4></value>
        </member>
        <member>
          <name>Page</name>
          <value><i4>1</i4></value>
        </member>
        <member>
          <name>Port</name>
          <value><i4>0</i4></value>
        </member>
      </struct>
    </member>
  </struct>
</member>
```

```
<name>PositionType</name>
  <value><string>key</string></value>
</member>
</struct></value>
</member>
<member>
  <name>CommandsToDelete</name>
  <value><array><data>
    <value><int>1426414447</int></value>
    <value><int>314528588</int></value>
    <value><int>1976270605</int></value>
    <value><int>584610977</int></value>
  </data></array></value>
</member>
</struct></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>
```

8.10.4.8 Object type 'call-to-conference'

8.10.4.8.1 Specific parameters for creating "call-to-conference"

The call to conference command has the following command specific parameters. The members AudioPriority, UseSecondChannel, Talk and Listen are optional. If they do not exist, RRCS takes default values.

Parameters	Type
Conference	<int> (object-ID of conference-object)
CommandPosition	<{TCmdPosition}>
UseSecondChannel	<boolean> (optional, default=false)
Talk	<boolean> (optional, default=true)
Listen	<boolean> (optional, default=true)
AudioPriority	<int> (optional, default = 1) - 0 = below standard - 1 = standard - 2 = high - 3 = paging - 4 = emergency

Example for creating a call to conference (Object ID 38476590) on Key 7 on physical port 5, Expansion Panel 3, Shift Page on node 2

```
<?xml version="1.0"?>
```

```
<methodCall>
```

```
<methodName>ConfigurationChange</methodName>
```

```
<params>
```

```
<param>
```

```
<value>C00000000001</value>
```

```
</param>
```

```
<param>
```

```
<value><array><data>
```

```
<value><struct>
```

```
<member>
```

```
<name>ChangeType</name>
```

```
<value>create</value>
```

```
</member>
```

```
<member>
```

```
<name>ObjectType</name>
```

```
<value>call-to-conference</value>
```

```
</member>
```

<member>

<name>SpecificParams</name>

<value><struct>

<member>

<name>CommandPosition</name>

<value><struct>

<member>

<name>ExpansionPanel</name>

<value><i4>3</i4></value>

</member>

<member>

<name>KeyNumber</name>

<value><i4>7</i4></value>

</member>

<member>

<name>Node</name>

<value><i4>2</i4></value>

</member>

<member>

<name>Page</name>

<value><i4>2</i4></value>

</member>

<member>

<name>Port</name>

<value><i4>5</i4></value>

</member>

<member>

<name>PositionType</name>

<value>key</value>

</member>

</struct></value>

</member>


```
<member>

  <name>Conference</name>

  <value><i4>38476590</i4></value>

</member>

</struct></value>

</member>

</struct></value>

</data></array></value>

</param>

</params>

</methodCall>
```

8.10.4.8.2 *Specific parameters for editing “call-to-conference”*

Parameters	Type
Conference	<int> (object-ID of conference-object)
CommandPosition	<{TCmdPosition}>
UseSecondChannel	<boolean> (default=false)
NewProperties	<struct> (contains the new properties of the call-to-conference-object, see table below)

NewProperties (all members are optional)	Property-type
Conference	<int> (object-ID of conference-object)
UseSecondChannel	<boolean>
Talk	<boolean>
Listen	<boolean>
Owner	<int> (object-ID of user-object)
AudioPriority	<int> <ul style="list-style-type: none"> - 0 = below standard - 1 = standard - 2 = high - 3 = paging - 4 = emergency

8.10.4.8.3 Example

Example for disabling the talk-privilege of the conference with object-ID=994082254 :

```
<?xml version="1.0"?>
```

```
<methodCall>
```

```
<methodName>ConfigurationChange</methodName>
```

```
<params>
```

```
<param>
```

```
<value>C0000000011</value>
```

```
</param>
```

```
<param>
```

```
<value><struct>
```

```
<member>
```

```
<name>ChangeType</name>
```

```
<value>edit</value>
```

</member>

<member>

<name>ObjectType</name>

<value>call-to-conference</value>

</member>

<member>

<name>SpecificParams</name>

<value><struct>

<member>

<name>CommandPosition</name>

<value><struct>

<member>

<name>KeyNumber</name>

<value><i4>7</i4></value>

</member>

<member>

```
<name>Node</name>

<value><i4>2</i4></value>

</member>

<member>

  <name>Page</name>

  <value><i4>1</i4></value>

</member>

<member>

  <name>Port</name>

  <value><i4>0</i4></value>

</member>

<member>

  <name>PositionType</name>

  <value>key</value>

</member>

</struct></value>
```

</member>

<member>

<name>Conference</name>

<value><i4>994082254</i4></value>

</member>

<member>

<name>NewProperties</name>

<value><struct>

<member>

<name>Talk</name>

<value><boolean>0</boolean></value>

</member>

</struct></value>

</member>

</struct></value>

</member>

</struct></value>

</param>

</params>

</methodCall>

8.10.4.8.4 *Specific parameters for deleting “call-to-conference”*

Hint: if the command does not exist, the delete operation is empty (no error on return)

<member>

<name>CommandPosition</name>

<value><{TCmdPosition}</value>

</member>

<member>

<name>Conference</name>

<value><int>{Object ID}</int></value>

</member>

<member>

<name>UseSecondChannel</name> <!-- Default: 0 -->

<value><boolean>{...}</boolean></value>

</member>

8.10.4.9 Object type 'call-to-group'

8.10.4.9.1 Specific parameters for creating "call-to-group"

Parameters	Type
Group	<int> (object-ID of group-object)
CommandPosition	<{TCmdPosition}>
UseSecondChannel	<boolean> (optional, default=false)
ShowIncomingMarker	<boolean> (optional, default=false)
DisableDestinVolumeAdjust	<boolean> (optional, default=false)
AudioPriority	<int> (optional, default = 1) <ul style="list-style-type: none"> - 0 = below standard - 1 = standard - 2 = high - 3 = paging - 4 = emergency

Example for creating a call to group (object ID 521994316) on the vox function on physical port 0 on node 2:

```
<?xml version="1.0"?>
```

```
<methodCall>
```

```
<methodName>ConfigurationChange</methodName>
```

```
<params>
```

```
<param>
```

<value>C00000000002</value>

</param>

<param>

<value><array><data>

<value><struct>

<member>

<name>ChangeType</name>

<value>create</value>

</member>

<member>

<name>ObjectType</name>

<value>call-to-group</value>

</member>

<member>

<name>SpecificParams</name>

<value><struct>

```
<member>

  <name>CommandPosition</name>

  <value><struct>

    <member>

      <name>IsInput</name>

      <value><boolean>1</boolean></value>

    </member>

  </member>

  <member>

    <name>Node</name>

    <value><i4>2</i4></value>

  </member>

  <member>

    <name>Port</name>

    <value><i4>0</i4></value>

  </member>

  <member>
```

<name>PositionType</name>

<value>virtual-function</value>

</member>

<member>

<name>VirtualFunctionType</name>

<value>vox</value>

</member>

</struct></value>

</member>

<member>

<name>Group</name>

<value><i4>521994316</i4></value>

</member>

</struct></value>

</member>

</struct></value>

```
</data></array></value>
```

```
</param>
```

```
</params>
```

```
</methodCall>
```

8.10.4.9.2 *Specific parameters for editing “call-to-group”*

Parameters	Type
Group	<int> (object-ID of group-object)
CommandPosition	<{TCmdPosition}>
UseSecondChannel	<boolean> (default=false)
NewProperties	<struct> (contains the new properties of the call-to-group-object, see table below)

NewProperties (all members are optional)	Property-type
Group	<int> (object-ID of group-object)
UseSecondChannel	<boolean>
Owner	<int> (object-ID of user-object)
AudioPriority	<int> <ul style="list-style-type: none"> - 0 = below standard - 1 = standard - 2 = high - 3 = paging - 4 = emergency
ShowIncomingMarker	<boolean> (optional, default=false)
DisableDestinVolumeAdjust	<boolean> (optional, default=false)

Example for enabling the incoming marker of the group with object-ID=521994316:

```
<?xml version="1.0"?>
```

```
<methodCall>
```

```
<methodName>ConfigurationChange</methodName>
```

```
<params>
```

```
<param>
```

<value>C00000000008</value>

</param>

<param>

<value><struct>

<member>

<name>ChangeType</name>

<value>edit</value>

</member>

<member>

<name>ObjectType</name>

<value>call-to-group</value>

</member>

<member>

<name>SpecificParams</name>

<value><struct>

<member>

<name>CommandPosition</name>

<value><struct>

<member>

<name>KeyNumber</name>

<value><i4>7</i4></value>

</member>

<member>

<name>Node</name>

<value><i4>2</i4></value>

</member>

<member>

<name>Page</name>

<value><i4>1</i4></value>

</member>

<member>

<name>Port</name>

<value><i4>0</i4></value>

</member>

<member>

<name>PositionType</name>

<value>key</value>

</member>

</struct></value>

</member>

<member>

<name>Group</name>

<value><i4>521994316</i4></value>

</member>

<member>

<name>NewProperties</name>

<value><struct>

<member>

```
<name>ShowIncomingMarker</name>

<value><boolean>1</boolean></value>

</member>

</struct></value>

</member>

</struct></value>

</member>

</struct></value>

</param>

</params>

</methodCall>
```

8.10.4.9.3 *Specific parameters for deleting “call-to-group”*

Hint: if the command does not exist, the delete operation is empty (no error on return)

```
<member>
```

```
<name>CommandPosition</name>
```

```
<value><{TCmdPosition}</value>
```

```
</member>
```

```
<member>
```

```
<name>Group</name>
```

```
<value><int>{Object ID}</int></value>
```

```
</member>
```

```
<member>
```

```
<name>UseSecondChannel</name> <!-- Default: 0 -->
```

```
<value><boolean>{...}</boolean ></value>
```

```
</member>
```

8.10.4.10 Object type 'call-to-ifb'

8.10.4.10.1 Specific parameters for creating "call-to-ifb"

Parameters	Type
IFBObjectID	<int> (object-ID of IFB-object)
CommandPosition	<{TCmdPosition}>
UseSecondChannel	<boolean> (optional, default=false)

8.10.4.10.2 Specific parameters for deleting "call-to-ifb"

Hint: if the command does not exist, the delete operation is empty (no error on return)

Parameters	Type
IFBObjectID	<int> (object-ID of IFB-object)
CommandPosition	<{TCmdPosition}>
UseSecondChannel	<boolean> (optional, default=false)

8.10.4.10.3 Example

```
<?xml version="1.0"?>
```

```
<methodCall>
```

```
<methodName>ConfigurationChange</methodName>
```

```
<params>

<param>

  <value>C00000000003</value>

</param>

<param>

  <value><array><data>

    <value><struct>

      <member>

        <name>ChangeType</name>

        <value>create</value>

      </member>

      <member>

        <name>ObjectType</name>

        <value>call-to-ifb</value>

      </member>

      <member>
```

<name>SpecificParams</name>

<value><struct>

<member>

<name>CommandPosition</name>

<value><struct>

<member>

<name>KeyNumber</name>

<value><i4>1</i4></value>

</member>

<member>

<name>Node</name>

<value><i4>2</i4></value>

</member>

<member>

<name>Page</name>

<value><i4>1</i4></value>

</member>

<member>

<name>Port</name>

<value><i4>0</i4></value>

</member>

<member>

<name>PositionType</name>

<value>key</value>

</member>

</struct></value>

</member>

<member>

<name>IFBObjectID</name>

<value><i4>1932025380</i4></value>

</member>

<member>

```
<name>UseSecondChannel</name>  
  
<value><boolean>0</boolean></value>  
  
</member>  
  
</struct></value>  
  
</member>  
  
</struct></value>  
  
</data></array></value>  
  
</param>  
  
</params>  
  
</methodCall>
```


8.10.4.11 Object type 'dim-level-command'

8.10.4.11.1 Specific parameters for creating "dim-level-command"

Parameters	Type
CommandPosition	<{TCmdPosition}>
Source	<{TPortAddress}>
Destination	<{TPortAddress}>
SourceUsesSecondChannel	<boolean> (optional, default=false)
DestinationUsesSecondChannel	<boolean> (optional, default=false)
DimValue	<int> 0: 0 [dB] (attenuation) 1: 3 [dB] (.. 2: 6 [dB] (.. 3: 9 [dB] (.. 4: 12 [dB] (.. 5: 18 [dB] (.. 6: 24 [dB] (.. 7: Mute (..)

Example to create a dim-level command on the „On-Call“ function on physical port 0, node 2. The command will dim/attenuate the crosspoint from port (node=2, port=26) to port (node=5, port=17) by 12 dB.

```
<?xml version="1.0"?>
```

```
<methodCall>
```

```
<methodName>ConfigurationChange</methodName>
```

```
<params>

<param>

  <value>C00000000002</value>

</param>

<param>

  <value><array><data>

    <value><struct>

      <member>

        <name>ChangeType</name>

        <value>create</value>

      </member>

      <member>

        <name>ObjectType</name>

        <value>dim-level-command</value>

      </member>

      <member>
```

<name>SpecificParams</name>

<value><struct>

<member>

<name>CommandPosition</name>

<value><struct>

<member>

<name>IsInput</name>

<value><boolean>0</boolean></value>

</member>

<member>

<name>Node</name>

<value><i4>2</i4></value>

</member>

<member>

<name>Port</name>

<value><i4>0</i4></value>

</member>

<member>

<name>PositionType</name>

<value>virtual-function</value>

</member>

<member>

<name>VirtualFunctionType</name>

<value>on-call</value>

</member>

</struct></value>

</member>

<member>

<name>Destination</name>

<value><struct>

<member>

<name>IsInput</name>

<value><boolean>0</boolean></value>

</member>

<member>

<name>Node</name>

<value><i4>5</i4></value>

</member>

<member>

<name>Port</name>

<value><i4>17</i4></value>

</member>

</struct></value>

</member>

<member>

<name>DimValue</name>

<value><i4>4</i4></value>

</member>

```
<member>

  <name>Source</name>

  <value><struct>

    <member>

      <name>IsInput</name>

      <value><boolean>1</boolean></value>

    </member>

  <member>

    <name>Node</name>

    <value><i4>2</i4></value>

  </member>

</member>

  <name>Port</name>

  <value><i4>26</i4></value>

</member>

</struct></value>
```

</member>

</struct></value>

</member>

</struct></value>

</data></array></value>

</param>

</params>

</methodCall>

8.10.4.11.2 *Specific parameters for editing “dim-level-command”*

Parameters	Type
CommandPosition	<{TCmdPosition}>
Source	<{TPortAddress}>
Destination	<{TPortAddress}>
SourceUsesSecondChannel	<boolean> (optional, default=false)
DestinationUsesSecondChannel	<boolean> (optional, default=false)
NewProperties	<struct> (contains the new properties of the dim-level-command -object, see

table below)

NewProperties (all members are optional)	Property-type
Source	<{TPortAddress}>
Destination	<{TPortAddress}>
SourceUsesSecondChannel	<boolean>
DestinationUsesSecondChannel	<boolean>
Source	<{TPortAddress}>
DimValue	<int> 0: 0 [dB] (attenuation) 1: 3 [dB] (.. 2: 6 [dB] (.. 3: 9 [dB] (.. 4: 12 [dB] (.. 5: 18 [dB] (.. 6: 24 [dB] (.. 7: Mute (..)

Example to change the dim-level 24 [dB] on the „On-Call“ function on physical port 0, node 2. The command will dim/attenuate the crosspoint from port (node=2, port=26) to port (node=5, port=17).

```
<?xml version="1.0"?>
```

```
<methodCall>
```



```
<methodName>ConfigurationChange</methodName>
```

```
<params>
```

```
<param>
```

```
<value>C00000000002</value>
```

```
</param>
```

```
<param>
```

```
<value><array><data>
```

```
<value><struct>
```

```
<member>
```

```
<name>ChangeType</name>
```

```
<value>edit</value>
```

```
</member>
```

```
<member>
```

```
<name>ObjectType</name>
```

```
<value>dim-level-command</value>
```

```
</member>
```

<member>

<name>SpecificParams</name>

<value><struct>

<member>

<name>CommandPosition</name>

<value><struct>

<member>

<name>IsInput</name>

<value><boolean>0</boolean></value>

</member>

<member>

<name>Node</name>

<value><i4>2</i4></value>

</member>

<member>

<name>Port</name>

<value><i4>0</i4></value>

</member>

<member>

<name>PositionType</name>

<value>virtual-function</value>

</member>

<member>

<name>VirtualFunctionType</name>

<value>on-call</value>

</member>

</struct></value>

</member>

<member>

<name>Source</name>

<value><struct>

<member>

<name>IsInput</name>

<value><boolean>1</boolean></value>

</member>

<member>

<name>Node</name>

<value><i4>2</i4></value>

</member>

<member>

<name>Port</name>

<value><i4>26</i4></value>

</member>

</struct></value>

</member>

<member>

<name>Destination</name>

<value><struct>

<member>

<name>IsInput</name>

<value><boolean>0</boolean></value>

</member>

<member>

<name>Node</name>

<value><i4>5</i4></value>

</member>

<member>

<name>Port</name>

<value><i4>17</i4></value>

</member>

</struct></value>

</member>

<member>

<name>NewProperties</name>

```
<value><struct>

  <member>

    <name>DimValue</name>

    <value><i4>4</i4></value>

  </member>

</struct></value>

</member>

</struct></value>

</member>

</struct></value>

</data></array></value>

</param>

</params>

</methodCall>
```

8.10.4.11.3 *Specific parameters for deleting “dim-level-command”*

Hint: if the command does not exist, the delete operation is empty (no error on return)

```
<member>

    <name>CommandPosition</name>

    <value><{TCmdPosition}</value>

</member>

<member>

    <name>Source</name>

    <value>{TPortAddress}</value>

</member>

<member>

    <!-- this member is optional, default value = false -->

    <name>SourceUsesSecondChannel</name>

    <value><boolean>..</boolean></value>

</member>

<member>

    <name>Destination</name>
```

```
<value>{TPortAddress}</value>
```

```
</member>
```

```
<member>
```

```
<!-- this member is optional, default value = false -->
```

```
<name>DestinationUsesSecondChannel</name>
```

```
<value><boolean>..</boolean></value>
```

```
</member>
```


8.10.4.12 Object type 'call-to-port-cmd'

8.10.4.12.1 Specific parameters for creating "call-to-port-cmd"

Parameters	Type
CommandPosition	<{TCmdPosition}>
SourceUsesSecondChannel	<boolean> (optional, default=false)
DestinationPortAddress	<{TPortAddress}>
DestinationUsesSecondChannel	<boolean> (optional, default = false)
TrunkingNetAddr	<int>(optional, for trunking)
TrunkingPortAddr	<int>(optional, for trunking)
New properties	<struct> (contains the new properties of the dim-level-command -object, see table below)

NewProperties (all members are optional)	Property-type
Priority	<int> (optional, default=1(standard))
TrunkcallPriority	<int> (optional, default=1(standard))
Isolate	<boolean> (optional, default=false)
IsolateSelf	<boolean> (optional, default = false)
DisableCrosspointVolume	<boolean> (optional, default = false)
BeepDestOnCall	<boolean> (optional, default = false)
AutolistenFromDest	<boolean>(optional, default = false)
AllowSetInOutGain	<boolean>(optional,default = true)

Example for creating a call to port on Key 7 on physical port 5, Expansion Panel 3, Shift Page on node 2

```
<?xml version="1.0"?>
```

```
<methodCall>
```

```
<methodName>ConfigurationChange</methodName>
```

```
<params>
```

```
<param>
```

```
<value>C00000000001</value>
```

```
</param>
```

```
<param>
```

```
<value><array><data>
```

```
<value><struct>
```

```
<member>
```

```
<name>ChangeType</name>
```

```
<value>create</value>
```

```
</member>
```

<member>

<name>ObjectType</name>

<value>call-to-port-cmd</value>

</member>

<member>

<name>SpecificParams</name>

<value><struct>

<member>

<name>CommandPosition</name>

<value><struct>

<member>

<name>ExpansionPanel</name>

<value><i4>3</i4></value>

</member>

<member>

<name>KeyNumber</name>

<value><i4>7</i4></value>

</member>

<member>

<name>Node</name>

<value><i4>2</i4></value>

</member>

<member>

<name>Page</name>

<value><i4>2</i4></value>

</member>

<member>

<name>Port</name>

<value><i4>5</i4></value>

</member>

<member>

<name>PositionType</name>

```
<value>key</value>

</member>

</struct></value>

</member>

<member>

  <name>SourceUsesSecondChannel</name>

  <value><boolean>0</boolean ></value>

</member>

<member>

  <name>DestinationPortAddress</name>

  <value><{TportAdress}></value>

</member>

<member>

  <name>DestinationUsesSecondChannel</name>

  <value><boolean>0</boolean></value>

</member>
```

<member>

<name>TrunkingNetAddr</name>

<value><i4>0</i4></value>

</member>

<member>

<name>TrunkingPortAddr</name>

<value><i4>0</i4></value>

</member>

<member>

<name>Priority</name>

<value><i4>1</i4></value>

</member>

<member>

<name>TrunkCallPriority</name>

<value><i4>1</i4></value>

</member>

<member>

<name>Isolate</name>

<value><boolean>0</boolean></value>

</member>

<member>

<name>IsolateSelf</name>

<value><boolean>0</boolean></value>

</member>

<member>

<name>DisableCrosspointVolume</name>

<value><boolean>0</boolean></value>

</member>

<member>

<name>BeepDestinationOnCall</name>

<value><boolean>0</boolean></value>

</member>

<member>

<name>AutolistenFromDest</name>

<value><boolean>0</boolean></value>

</member>

<member>

<name>AllowSetInOutGain</name>

<value><boolean>0</boolean></value>

</member>

</struct></value>

</member>

</struct></value>

</data></array></value>

</param>

</params>

</methodCall>

8.10.4.12.2 *Specific parameters for deleting “call-to-port-cmd”*

Hint: if the command does not exist, the delete operation is empty (no error on return)

Parameters	Type
CommandPosition	<{TCmdPosition}>
SourceUsesSecondChannel	<boolean> (optional, default=false)
DestinationPortAddress	<{TPortAddress}>
DestinationUsesSecondChannnel	<boolean> (optional, default = false)
TrunkingNetAddr	<int>(optional, for trunking)
TrunkingPortAddr	<int>(optional, for trunking)

Example for deleting a call to port on Key 7 on physical port 5, Expansion Panel 3, Shift Page on node 2

```
<?xml version="1.0"?>
```

```
<methodCall>
```

```
<methodName>ConfigurationChange</methodName>
```

```
<params>
```

```
<param>

  <value>C00000000001</value>

</param>

<param>

  <value><array><data>

    <value><struct>

      <member>

        <name>ChangeType</name>

        <value>delete</value>

      </member>

      <member>

        <name>ObjectType</name>

        <value>call-to-port-cmd</value>

      </member>

      <member>

        <name>SpecificParams</name>
```

<value><struct>

<member>

<name>CommandPosition</name>

<value><struct>

<member>

<name>ExpansionPanel</name>

<value><i4>3</i4></value>

</member>

<member>

<name>KeyNumber</name>

<value><i4>7</i4></value>

</member>

<member>

<name>Node</name>

<value><i4>2</i4></value>

</member>

<member>

<name>Page</name>

<value><i4>2</i4></value>

</member>

<member>

<name>Port</name>

<value><i4>5</i4></value>

</member>

<member>

<name>PositionType</name>

<value>key</value>

</member>

</struct></value>

</member>

<member>

<name>SourceUsesSecondChannel</name>

```
<value><boolean>0</boolean></value>

</member>

<member>

  <name>DestinationPortAddress</name>

  <value><{TportAdress}></value>

</member>

<member>

  <name>DestinationUsesSecondChannel</name>

  <value><boolean>0</boolean></value>

</member>

<member>

  <name>TrunkingNetAddr</name>

  <value><i4>0</i4></value>

</member>

<member>

  <name>TrunkingPortAddr</name>
```

<value><i4>0</i4></value>

</member>

</struct></value>

</member>

</struct></value>

</data></array></value>

</param>

</params>

</methodCall>

8.10.4.13 Object type 'listen-to-port-cmd'

8.10.4.13.1 Specific parameters for creating "listen-to-port-cmd"

Parameter	Type
CommandPosition	<{TCmdPosition}>
LocalPortAddress	<{TPortAddress}>
SourceUsesSecondChannel	<boolean> (optional, default=false)
LocalUsesSecondChannel	<boolean> (optional, default = false)
TrunkingNetAddr	<int>(optional, for trunking)
TrunkingPortAddr	<int>(optional, for trunking)
New properties	<struct> (contains the new properties of the dim-level-command -object, see table below)

NewProperties (all members are optional)	Property-type
Priority	<int> (optional, default=1(standard))
TrunkcallPriority	<int> (optional, default=1(standard))
DisableCrosspointVolume	<boolean> (optional, default=false)
AllowSetInOutGain	<boolean> (optional, default=false)

8.10.4.13.2 Specific parameters for deleting "listen-to-port-cmd"

Parameters	Type
CommandPosition	<{TCmdPosition}>

LocalPortAddress	<{TPortAddress}>
SourceUsesSecondChannel	<boolean> (optional, default=false)
LocalUsesSecondChannel	<boolean> (optional, default = false)
TrunkingNetAddr	<int>(optional, for trunking)
TrunkingPortAddr	<int>(optional, for trunking)

8.10.4.14 Object type “route-cmd”

8.10.4.14.1 Specific parameters for creating “route-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
SourcePortAddress	<{TPortAddress}>
SourceUsesSecondChannel	<boolean> (optional, default = false)
DestinationPortAddress	<{TPortAddress}>
DestinationUsesSecondChannel	<boolean> (optional, default = false)
New properties	<struct> (contains the new properties of the dim-level-command -object, see table below)

NewProperties (all members are optional)	Property-type
Priority	<int> (optional, default=1(standard))
DisableCrosspointVolume	<boolean> (optional, default=false)

8.10.4.14.2 Specific parameters for deleting “route-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
SourcePortAddress	<{TPortAddress}>

SourceUsesSecondChannel	<boolean> (optional, default = false)
DestinationPortAddress	<{TPortAddress}>

8.10.4.15 Object type “switch-gpio-cmd”

8.10.4.15.1 Specific parameters for creating “switch-gpio-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
PortAddress	<{TPortAddress}>
Slot	<int>
GPIONumber	<int>

This object type hasn't got any properties.

8.10.4.15.2 Specific parameters for deleting “switch-gpio-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
PortAddress	<{TPortAddress}>
Slot	<int>
GPIONumber	<int>

8.10.4.16 Object type “select-audiopatch-cmd”

8.10.4.16.1 Specific parameters for creating “select-audiopatch-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
PortAddress	<{TPortAddress}>
Audiopatch	<int>

This object type hasn't got any properties.

8.10.4.16.2 Specific parameters for deleting “select-audiopatch-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
PortAddress	<{TPortAddress}>
Audiopatch	<int>

8.10.4.17 Object type “control-audiopatch-cmd”

8.10.4.17.1 Specific parameters for creating “control-audiopatch-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
PortAddress	<{TPortAddress}>
AudiopatchElement	<int>

This object type hasn't got any properties.

8.10.4.17.2 Specific parameters for deleting “control-audiopatch-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
PortAddress	<{TPortAddress}>
AudiopatchElement	<int>

8.10.4.18 Object type “remote-key-cmd”

8.10.4.18.1 Specific parameters for creating “remote-key-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
DestinationPortAddress	<{TPortAddress}>
ExpansionPanel	<int> (optional)
KeyPage	<int> 0 – standard 1 – shift page 2 – virtual key
KeyNumber	<int>
New properties	<struct> (contains the new properties of the dim-level-command -object, see table below)

NewProperties (all members are optional)	Property-type
PressKey	<boolean> (default = false)
LockKey	<boolean> (default = false)
MarkerFlag	<boolean> (default = false)
MarkerValue	<int> (default=0)
TextFlag	<boolean> (default = false)
TextValue	<string> (default=“”)

8.10.4.18.2 *Specific parameters for deleting “remote-key-cmd”*

Parameters	Type
CommandPosition	<{TCmdPosition}>
DestinationPortAddress	<{TPortAddress}>
ExpansionPanel	<int> (optional)
KeyPage	<int> 0 – standard 1 – shift page 2 – virtual key
KeyNumber	<int>

8.10.4.19 Object type “reply-cmd”

8.10.4.19.1 Specific parameters for creating “reply-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
New properties	<struct> (contains the new properties of the dim-level-command -object, see table below)
NewProperties (all members are optional)	Property-type
Priority	<int> (default=1(standard))
EnableForCallsFromConf	<boolean> (default = false)

8.10.4.19.2 Specific parameters for deleting “reply-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>

8.10.4.20 Object type “edit-conference-cmd”

8.10.4.20.1 Specific parameters for creating “edit-conference-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>

This object type hasn't got any properties.

8.10.4.20.2 ***Specific parameters for deleting “edit-conference-cmd”***

Parameters	Type
CommandPosition	<{TCmdPosition}>

8.10.4.21 Object type “edit-ifb-cmd”

8.10.4.21.1 Specific parameters for creating “edit-ifb-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>

This object type hasn't got any properties.

8.10.4.21.2 Specific parameters for deleting “edit-ifb-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>

8.10.4.22 Object type “dim-panel-speaker-cmd”

8.10.4.22.1 Specific parameters for creating “dim-panel-speaker -cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
DestinationPortAddress	<{TPortAddress}>
New properties	<struct> (contains the new properties of the dim-level-command -object, see table below)
NewProperties (all members are optional)	Property-type
DimSpeakerBy	<int> (default = 0)

8.10.4.22.2 Specific parameters for deleting “dim-panel-speaker-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
DestinationPortAddress	<{TPortAddress}>

8.10.4.23 Object type “beep-panel-cmd”

8.10.4.23.1 Specific parameters for creating “beep-panel-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
DestinationPortAddress	<{TPortAddress}>

This object type hasn't got any properties.

8.10.4.23.2 Specific parameters for deleting “beep-panel-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
DestinationPortAddress	<{TPortAddress}>

8.10.4.24 Object type “telephone-dial-keypad-cmd”

8.10.4.24.1 Specific parameters for creating “telephone-dial-keypad-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
KeyFunction	<int>

This object type hasn’t got any properties.

8.10.4.24.2 Specific parameters for deleting “telephone-dial-keypad-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
KeyFunction	<int>

8.10.4.24.3 *Specific parameters for editing “telephone-dial-keypad-cmd”*

NewProperties	Property-type
KeyFunction	<int>
KeyPhoneNumber	<string>
(Only available if KeyFunction = 17)	

8.10.4.25 Object type “telephone-dial-hangup-cmd”

8.10.4.25.1 Specific parameters for creating “telephone-dial-hangup-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
Function	<int>

This object type hasn't got any properties.

8.10.4.25.2 Specific parameters for deleting “telephone-dial-hangup-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
Function	<int>

8.10.4.26 Object type “logic-cmd”

8.10.4.26.1 Specific parameters for creating “logic-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
LogicSource	<int>

This object type hasn't got any properties.

8.10.4.26.2 Specific parameters for deleting “logic-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
LogicSource	<int>

8.10.4.27 Object type “kill-partyline-mic-cmd”

8.10.4.27.1 Specific parameters for creating “kill-partyline-mic-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>

This object type hasn't got any properties.

8.10.4.27.2 Specific parameters for deleting “kill-partyline-mic-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>

8.10.4.28 Object type “auto-listen-off-cmd”

8.10.4.28.1 Specific parameters for creating “auto-listen-off-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>

This object type hasn't got any properties.

8.10.4.28.2 Specific parameters for deleting “auto-listen-off-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>

8.10.4.29 Object type “set-io-gain-cmd”

8.10.4.29.1 Specific parameters for creating “set-io-gain-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
DestinationPortAddress	<{TPortAddress}>
IsInput	<boolean>

This object type hasn’t got any properties.

8.10.4.29.2 Specific parameters for deleting “set-io-gain-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
DestinationPortAddress	<{TPortAddress}>
IsInput	<boolean>

8.10.4.30 Object type “sidetone-cmd”

8.10.4.30.1 Specific parameters for creating “sidetone-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
New properties	<struct> (contains the new properties of the dim-level-command -object, see table below)
NewProperties (all members are optional)	Property-type
EnableForSpeakerMode	<boolean> (default = false)
EnableForHeadsetMode	<boolean> (default = false)
NormSidetoneLevel	<int> (default = -6)

8.10.4.30.2 Specific parameters for deleting “sidetone-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>

8.10.4.31 Object type “send-string-cmd”

8.10.4.31.1 Specific parameters for creating “send-string-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
SendString	<string>

This object type hasn't got any properties.

8.10.4.31.2 Specific parameters for deleting “send-string-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>
SendString	<string>

8.10.4.32 Object type “clone-output-port-cmd”

8.10.4.32.1 Specific parameters for creating “clone-output-port-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>

NewProperties (all members are optional)	Property-type
OutputToCloneAddress	<{TPortAddress}>
OutputToCloneSecondChannel	<boolean> (optional, default=false)
ClonedOutputAddress	<{TPortAddress}>
ClonedOutputSecondChannel	<boolean> (optional, default=false)

8.10.4.32.2 Specific parameters for deleting “clone-output-port-cmd”

Parameters	Type
CommandPosition	<{TCmdPosition}>

8.10.4.33 Object type “panel-key”

8.10.4.33.1 Specific parameters for editing “panel-key”

Parameters	Type
Node	<int>
Port	<int>
ExpansionPanel	<int>
Page	<int>
KeyNumber	<int>
IsInput	<boolean>
New properties	<struct> (contains the new properties of the dim-level-command -object, see table below)

NewProperties (all members are optional)	Property-type
AutoLabelFlag	<boolean> (optional, default=true)
LabelValue	<string> (optional, default=“”)
KeyMode	<int> (optional, default=1) 0 – Auto 1 – Momentary(PTT) 2 - Latching
LatchingTimeout	<int> (optional, default = 0)

	0 – Net Default 1 – Permanent 2 - 1 sec
RadioButton	<int> (optional, default = 0) 0 - <none> 1 – Group 1 2 – Group 2 3 – Group 3 4 - Group 4
Dim	<boolean> (optional, default = true)
ActionByKeyPressed	<int>(optional, default = 1) 0 – No Unmute when activated, no mute when deactivated 1 – Unmute when activated, no mute when deactivated 2 – Unmute when activated, mute again when deactivated
ScrollEnable	<boolean>(optional,default = false)
RestoreVolumeLevel	<boolean>(optional, default = false)

8.10.4.34 Object type “client-card”

8.10.4.34.1 Specific parameters for editing “client-card”

Parameters	Type
Node	<int>
Slot	<int>
VoIP	<struct>(contains the new properties of the VoIP properties, see table below)
Aes67	<struct>(contains the new properties of the AES67 properties, see table below)

VoIP (all members are optional)	Property-type
ObtainIpAddrAutomatic	<boolean>(optional)
ObtainDnsAddrAutomatic	<boolean> (optional)
IpAddress	{IP-address} (optional)
NetMask	<string> (optional)
DefaultGateway	{IP-address} (optional)
PrimaryDnsServer	{IP-address} (optional)
SecondaryDnsServer	{IP-address} (optional)
DnsHostName	<string> (optional)
TcpUdpPort	<int> (optional)
ServiceCodePoint	<int> (optional)
EthernetLinkMode	{EthernetLinkMode} (optional) 1 = Autonegotiation 2 = 10 Mbit Half

3 = 10 Mbit Full
4 = 100 Mbit Half
5 = 100 Mbit Full

AES67 (all members are optional)	Property-type
IpAddress	{IP-address} (optional)
Network	<string> (optional)
DefaultGateway	{IP-address} (optional)
TcpUdpPort	<int> (optional)
ServiceCodePoint	<int> (optional)
PTP	<int> (optional)
PtpPriority	<int> (optional)
PtpMode	<int> (optional) 0 = Multicast 1 = Hybrid

8.11 Key and marker manipulation

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
Clears a key label that has been set before	ClearKeyLabel	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><boolean>{IsInput}</boolean></value></param></code> <code><param><value><int>{page}</int></value></param></code> <code><param><value><int>{ExpansionPanel}</int></value></param></code> <code><param><value><int>{KeyNumber}</int></value></param></code> <code><param><value><boolean>{IsVirtKey}</boolean></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Clears a key label and marker that has been set before	ClearKeyLabelAndMarker	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><boolean>{IsInput}</boolean></value></param></code> <code><param><value><int>{page}</int></value></param></code> <code><param><value><int>{ExpansionPanel}</int></value></param></code> <code><param><value><int>{KeyNumber}</int></value></param></code> <code><param><value><boolean>{IsVirtKey}</boolean></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Clears a marker that has been set before	ClearKeyMarker	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><boolean>{IsInput}</boolean></value></param></code> <code><param><value><int>{page}</int></value></param></code> <code><param><value><int>{ExpansionPanel}</int></value></param></code> <code><param><value><int>{KeyNumber}</int></value></param></code> <code><param><value><boolean>{IsVirtKey}</boolean></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Queries all key manipulations that have	GetAllRemoteKeys	<code><param><value><string>{TransKey}</string></value></param></code>	<code><param><value><struct></code> <code><member></code>

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
been sent via RRCS with one the functions in this section			<pre> <name>RemoteKey</name> <value><array><data> <value><struct> <member> <name>EventId</name> <value><i4>{ EventId }</i4></value> </member> <member> <name>KeyNumer</name> <value><i4>{ KeyNumer }</i4></value> </member> <member> <name>Label</name> <value><string></string></value> </member> <member> <name>LabelFlag</name> <value> <boolean>{ LabelFlag }</boolean> </value> </member> <member> <name>LockKeyFlag</name> <value> <boolean>{ LockFlag }</boolean> </value> </member> <member> <name>Marker</name> <value><i4>{ Marker }</i4></value> </member> <member> <name>MarkerFlag</name> <value> <boolean>{ Marker }</boolean> </pre>

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
			<pre> </value> </member> <member> <name>Node</name> <value><i4>{ Node }</i4></value> </member> <member> <name>PanelId</name> <value><i4>{ PanelId }</i4></value> </member> <member> <name>PressKeyFlag</name> <value> <boolean>{ PressFlag }</boolean> </value> </member> <member> <name>SubPanel</name> <value> <i4>{SubPanel}</i4> </value> </member> </struct></value> ... </data></array></value> </member> <member> <name>Transkey</name> <value> <string>{Transkey}</string> </value> </member> </struct></value></param> </pre>

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
Queries key manipulations for the given key	GetRemoteKey	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><boolean>{IsInput}</boolean></value></param></code> <code><param><value><int>{page}</int></value></param></code> <code><param><value><int>{ExpansionPanel}</int></value></param></code> <code><param><value><int>{KeyNumber}</int></value></param></code> <code><param><value><boolean>{IsVirtKey}</boolean></value></param></code>	Same as GetAllRemoteKeys
Locks a key, so the user cannot press it.	LockKey	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><boolean>{IsInput}</boolean></value></param></code> <code><param><value><int>{page}</int></value></param></code> <code><param><value><int>{ExpansionPanel}</int></value></param></code> <code><param><value><int>{KeyNumber}</int></value></param></code> <code><param><value><boolean>{IsVirtKey}</boolean></value></param></code> <code><param><value><boolean>{Lock}</boolean></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Presses a key	PressKey	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><boolean>{IsInput}</boolean></value></param></code> <code><param><value><int>{page}</int></value></param></code> <code><param><value><int>{ExpansionPanel}</int></value></param></code> <code><param><value><int>{KeyNumber}</int></value></param></code> <code><param><value><boolean>{IsVirtKey}</boolean></value></param></code> <code><param><value><boolean>{Press}</boolean></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Sets a key label. It overwrites any other key label of the configuration. {Label} is a 8 character	SetKeyLabel	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><boolean>{IsInput}</boolean></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
string.		<code><param><value><int>{page}</int></value></param></code> <code><param><value><int>{ExpansionPanel}</int></value></param></code> <code><param><value><int>{KeyNumber}</int></value></param></code> <code><param><value><boolean>{IsVirtKey}</boolean></value></param></code> <code><param><value><string>{Label}</string></value></param></code>	
<p>Sets a key label and a marker .</p> <p>{Label} is a 8 character string.</p> <p>{Marker} is the index of the marker in the configuration. See the net properties dialog in Director.</p>	SetKeyLabelAndMarker	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><boolean>{IsInput}</boolean></value></param></code> <code><param><value><int>{page}</int></value></param></code> <code><param><value><int>{ExpansionPanel}</int></value></param></code> <code><param><value><int>{KeyNumber}</int></value></param></code> <code><param><value><boolean>{IsVirtKey}</boolean></value></param></code> <code><param><value><string>{Label}</string></value></param></code> <code><param><value><int>{Marker}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
<p>Sets a marker above the key.</p> <p>{Marker} is the index of the marker in the configuration. See the net properties dialog in Director.</p>	SetKeyMarker	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><boolean>{IsInput}</boolean></value></param></code> <code><param><value><int>{page}</int></value></param></code> <code><param><value><int>{ExpansionPanel}</int></value></param></code> <code><param><value><int>{KeyNumber}</int></value></param></code> <code><param><value><boolean>{IsVirtKey}</boolean></value></param></code> <code><param><value><int>{Marker}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>

8.12 Panel-Spy

Description	Remote Procedure	Parameters	Return Value on success
Starts/Stops panel spy notifications. Hint: The method fails, if RegisterForAllEvents has not been called.	ChangePanelSpyRegistry	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{TCPPort}</int></value></param></code> <code><param><value><string>{URLPath}</string></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><struct>{EventInfo}</struct></value></param></code>	<code><param><value><string>{TransKey}</string></value></param></code> <code></value></code>

Input parameters

Name	Type	Parameter description
TransKey	string	<ul style="list-style-type: none"> Transaction Key
TCPPort	int	<ul style="list-style-type: none"> TCP Port
URLPath	string	<p>Defines the URL-path of the receiving computer. E.g. if path is 'notification' RRCS sends the events to the URL: "http://[ip-address of calling computer]:[TCPPort]/notification". String conditions:</p> <ul style="list-style-type: none"> All characters must be ASCII characters excluding the control characters ([0..31,127]) and the space character. The leading character can be a slash however it needs not <p>default: <code><value>/RPC2</value></code></p>
Node	int	<ul style="list-style-type: none"> Node address

Port	int	<ul style="list-style-type: none"> • Port address
EventInfo	struct	<p>The struct can contain the following members:</p> <ul style="list-style-type: none"> • RotateEventsOn (type bool) • KeyEventsOn (type bool) • FuncKeyEventsOn (type bool) • NumKeyEventsOn (type bool) <ul style="list-style-type: none"> - if not present, the notification feature is nor turned on neither turned off. - if present, the notification feature is turned on (struct member = true) or turned off (struct member = false).

8.13 Port Cloning

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
Clones the output signal of a port and routes all audios to a monitoring port.	StartPortCloning	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value>{Port to monitor}</value></param></code> <code><param><value>{Clone Port}</value></param></code>	<code><param></code> <code> <value></code> <code> <string>{TransKey}</string></code> <code> </value></code> <code></param></code>
Stops cloneing. Use Port 0.0 as a wildcard in order to stop a batch of port clones.	StopPortCloning	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value>{Port to monitor}</value></param></code> <code><param><value>{Clone Port}</value></param></code>	<code><param></code> <code> <value></code> <code> <string>{TransKey}</string></code> <code> </value></code> <code></param></code>
Returns an array of all active port clones.	GetAllActivePort Clones	<code><param><value><string>{TransKey}</string></value></param></code>	<code><param></code> <code> <value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><array><data></code> <code> <!--1st port clone information --></code> <code> <value><struct></code> <code> <member></code> <code> <name>PortToMonitor</name></code> <code> <value></code> <code> <struct>{Port to monitor}</struct></code> <code> </value></code> <code> </member></code> <code> <member></code> <code> <name>ClonePort</name></code> <code> <value></code> <code> <struct>{Clone Port}<struct></code> <code> </value></code> <code> </member></code> <code></value></code> <code></param></code>

Description	Remote Procedure	Parameters	Return Value on success (else XML-RPC-Fault-Response)
			... </struct></value> <!--further port clones --> ... </data></array></value> </data></array></value> </param> </params>

{Port to monitor} and {Clone Port} are using the existing helper type <TPortAddress>.

8.14 IFB Volume

Description	Remote Procedure	Parameters	Return Value on success
<p>Sets the mix minus volume for an IFB. It affects the single volume.</p> <p>If the mix minus and the output is assigned when SetIFBVolumeMixMinus is called, the volume for the crosspoint MixMinus->Output is set.</p> <p>If SetIFBVolumeMixMinus is called and a mix minus is not assigned, it will return an error. Presetting the volume is therefor not possible.</p> <p>If SetIFBVolumeMixMinus is called and an output is not assigned, the value is stored.</p>	SetVolumeIFBMixMinus	<pre> <param><value> <string>{TransKey}</string> </value></param> <param><value> <int>{Node}</int> </value></param> <param><value> <int>{Port}</int> </value></param> <param><value> <boolean>{IsInput}</boolean> </value></param> <param><value> <int>{IfbNumber}</int> </value></param> <param><value> <int>{Volume}</int> </value></param> </pre>	<pre> <param> <value> <string>{TransKey}</string> </value> </param> </pre>

Description	Remote Procedure	Parameters	Return Value on success
<p>Later when the output is assigned the crosspoint volume is set to the IFB volume. If an output is reassigned, it does not affect the IFB volume.</p> <p>When the mix minus is assigned the initial volume is always unity gain.</p> <p>If a mix minus is a group, the volumes for the group members can be set separately and do not interfere with each other.</p>			
<p>Returns the the mix minus volume for an IFB</p> <p>If the volume was never set by SetIFBVolumeMixMinus, the return value is unity gain.</p>	GetIFBVolumeMixMinus	<pre> <param><value> <string>{TransKey}</string> </value></param> <param><value> <int>{Node}</int> </value></param> <param><value> <int>{Port}</int> </value></param> <param><value> <boolean>{IsInput}</boolean> </value></param> <param><value> </pre>	<pre> <param><value> <array><data> <value><string>{TransKey}</string></value> <value><array><data> <value><struct> <member> <name>IfbNumber</name> <value><int>{IfbNumber}</int></value> </member> <member> <name>Node</name> <value><int>{Node}</int></value> </member> </pre>

Description	Remote Procedure	Parameters	Return Value on success
		<pre> <int>{IfbNumber}</int> </value></param> </pre>	<pre> <member> <name>Port</name> <value><int>{Port}</int></value> </member> <member> <name>Volume</name> <value><int>{Volume}</int></value> </member> </struct></value> </data></array></value> </data></array> </value></param> </pre>
Removes the mix minus volume for an IFB. The volume returns to default (untiy gain)	RemoveVolumeIFBMixMinus	<pre> <param><value> <string>{TransKey}</string> </value></param> <param><value> <int>{Node}</int> </value></param> <param><value> <int>{Port}</int> </value></param> <param><value> <boolean>{IsInput}</boolean> </value></param> <param><value> <int>{IfbNumber}</int> </value></param> <param><value> <int>{Volume}</int> </value></param> </pre>	<pre> <param> <value> <string>{TransKey}</string> </value> </param> </pre>

8.15 Registration for notifications

Description	Remote Procedure	Parameters	Return Value on success
<p>Registers volume change notifications for the given crosspoints.</p> <p>Hint:</p> <ul style="list-style-type: none"> - RRCS will send initial volume values for all crosspoints. - The method fails, if RegisterForEventsEx has not been called with the member "XpVolumeChange" set to true - The method fails, if the total number of registered crosspoints would exceed the limit of 8192 crosspoints. - If the method fails, the registry keeps unchanged 	XpVolumeChangeRegistryReset	<pre> <param> <value><string>{TransKey}</string></value> </param> <param> <value><string>{IP-address}</string></value> </param> <param> <value><i4>{TCP-port}</i4></value> </param> <param> <value><array><data> <!-- 1st crosspoint --> <value><struct> <member> <name>Destination</name> <value><struct> <member> <name>IsInput</name> <value><boolean>0</boolean></value> </member> <member> <name>Node</name> <value><i4>{Node}</i4></value> </member> <member> <name>Port</name> <value><i4>{Port}</i4></value> </member> </struct></value> </member> </struct></value> </param> </pre>	<pre> <param><value><string>{TransKey}</string></value> </pre>

Description	Remote Procedure	Parameters	Return Value on success
		<pre> <member> <name>Source</name> <value><struct> <member> <name>IsInput</name> <value><boolean>1</boolean></value> </member> <member> <name>Node</name> <value><i4>{Node}</i4></value> </member> <member> <name>Port</name> <value><i4>{Port}</i4></value> </member> </struct></value> </member> </struct></value> <!-- further crosspoints --> ... </param> </pre>	
<p>Adds crosspoints to the volume change notification registry.</p> <p>Hint:</p> <ul style="list-style-type: none"> - RRCS will send initial volume values for all newly registered crosspoints. - The method fails, if RegisterForEventsEx has not been called 	XpVolumeChangeRegistryAdd	See method "XpVolumeChangeRegistryReset"	<param><value><string>{TransKey}</string></value>

Description	Remote Procedure	Parameters	Return Value on success
with the member "XpVolumeChange" set to true - The method fails, if the total number of registered crosspoints would exceed the limit of 8192 crosspoints. - If the method fails, the registry keeps unchanged			
Removes a set of crosspoints from the volume change registry - The method fails, if RegisterForEventsEx has not been called with the member "XpVolumeChange" set to true. In this case nothing will be removed from the registry.	XpVolumeChangeRegistryRemove	See method "XpVolumeChangeRegistryReset"	<param><value><string>{TransKey}</string></value>
Register Control System client for events	RegisterForEvents	<param><value><string>{TransKey}</string></value></param> <param><value><string>{IP-address}</string></value></param> <param><value><int>{TCP-Port}</int></value></param>	<param><value><array><data> <value><string>{TransKey}</string></value> <value><int>{ErrorCode}</int></value> </data></array></value></param>
Unregister Control System client for events	UnregisterForEvents	<param><value><string>{TransKey}</string></value></param> <param><value><string>{IP-address}</string></value></param>	<param><value><array><data> <value><string>{TransKey}</string></value> <value><int>{ErrorCode}</int></value> </data></array></value></param>

Description	Remote Procedure	Parameters	Return Value on success
Registration for GP input change notifications	RegisterForGpIn putChange	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><string>{IP-address}</string></value></param></code> <code><param><value><int>{TCP-Port}</int></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Unregistration for GP input change notifications	UnregisterForGp InputChange	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><string>{IP-address}</string></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Registration for GP output change notifications	RegisterForGpO utputChange	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><string>{IP-address}</string></value></param></code> <code><param><value><int>{TCP-Port}</int></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Unregistration for GP output change notifications	UnregisterForGp OutputChange	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><string>{IP-address}</string></value></param></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Registration for change notifications Hint: The struct-members: General GpInChange GpOutChange XpChange LogicSourceChange XpVolumeChange ConfigurationChange are optional. If they are not given, no notifications will be	RegisterForEve ntsEx	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><string>{IP-address}</string></value></param></code> <code><param><value><int>{TCP-Port}</int></value></param></code> <code><param><value><struct></code> <code> <member></code> <code> <name>General</name></code> <code> <value><boolean>{On/Off}</boolean></value></code> <code> <member></code> <code> <name>GpInChange</name></code> <code> <value><boolean>{On/Off}</boolean></value></code> <code></member></code> <code><member></code> <code> <name>GpOutChange</name></code>	<code><param><value><array><data></code> <code> <value><string>{TransKey}</string></value></code> <code> <value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>

Description	Remote Procedure	Parameters	Return Value on success
sent.		<pre> <value><boolean>{On/Off}</Boolean></value> </member> </member> <member> <name>XpChange</name> <value><boolean>{On/Off}</boolean></value> </member> </member> <member> <!--requires RRCS v5.91 or later --> <name>LogicSourceChange</name> <value><boolean>{On/Off}</boolean></value> </member> <member> <!--requires RRCS v5.91 or later --> <!-- see methods "XpVolumeChangeRegistryReset", "XpVolumeChangeRegistryAdd" and "XpVolumeChangeRegistryRemove" --> <name>XpVolumeChange</name> <value><boolean>{On/Off}</boolean></value> </member> </struct></value></param> </pre>	
Unregistration for change notifications	UnregisterForEventsEx	<pre> <param><value><string>{TransKey}</string></value></param> <param><value><string>{IP-address}</string></value></param> </pre>	<pre> <param><value><array><data> <value><string>{TransKey}</string></value> <value><int>{ErrorCode}</int></value> </data></array></value></param> </pre>

8.15.1 RegisterForAllEvents

Input parameters

Name	Type	Parameter description
TransKey	string	<ul style="list-style-type: none"> Transaction Key <p>default: <value>C0123456789</value></p>
TCPPort	int	<ul style="list-style-type: none"> TCP-Port <p>default: <value><i4>80</i4></value></p>
URLPath	string	<p>Defines the URL-path of the receiving computer. E.g. if path is 'notification' RRCS sends the events to the URL: "http://[ip-address of calling computer]:[TCPPort]/notification". String conditions:</p> <ul style="list-style-type: none"> All characters must be ASCII characters excluding the control characters ([0..31,127]) and the space character. The leading character can be a slash however it needs not <p>default: <value>/RPC2</value></p>
AllowHttpPipelining	boolean	<ul style="list-style-type: none"> 0 (==false): Wait for acknowledge between outgoing notifications (slow, extra network round trips) 1 (==true): RRCS sends multiple HTTP-packets without waiting for each response <p>default: <value><boolean>0</boolean></value></p>

AllowSystemMulticall	boolean	<p>NOTE: RRCS does not support the 'system.multicall' yet. However it may support it in future releases.</p> <ul style="list-style-type: none"> 0 (==false): Forbids RRCS to boxcar multiple RPC notifications in one request. 1 (==true): Allows RRCS to boxcar multiple RPC notifications in one request. <p>default: <value><boolean>0</boolean></value></p>
----------------------	---------	--

Description	Remote Procedure	Parameters	Return Value on success
Allows to receive any RRCS event. RRCS will send events to calling the computer with the given TCP-port.	RegisterForAllEvents	<param><value><string>{TransKey}</string></value></param> <param><value><int>{TCPPort}</int></value></param> <param><value><string>{URLPath}</string></value></param> <param><value><boolean>{AllowHttpPipelining}</boolean></value></param> <param><value><boolean>{AllowSystemMulticall}</boolean></value></param>	<param><value><string>{TransKey}</string></value></param>

8.15.2 UnregisterForAllEvents

Input parameters

Name	Type	Parameter description
TransKey	string	<ul style="list-style-type: none"> Transaction Key <p>default: <value>C0123456789</value></p>
TCPPort	int	<ul style="list-style-type: none"> TCP-Port <p>default: <value><i4>80</i4></value></p>
URLPath	string	<p>Defines the URL-path of the receiving computer. E.g. if path is 'notification' RRCS sends the events to the URL: "http://[ip-address of calling computer]:[TCPPort]/notification". String conditions:</p> <ul style="list-style-type: none"> All characters must be ASCII characters excluding the control characters ([0..31,127]) and the space character. The leading character can be a slash however it needs not <p>default: <value>/RPC2</value></p>

Description	Remote Procedure	Parameters	Return Value on success
Unregisters a single event receiver for all change notifications	UnregisterForAllEvents	<param><value><string>{TransKey}</string></value></param> <param><value><int>{TCPPort}</int></value></param> <param><value><string>{URLPath}</string></value></param>	<param><value><string>{TransKey}</string></value></param>

Description	Remote Procedure	Parameters	Return Value on success

8.16 Trunking

Description	Remote Procedure	Parameters	Return Value on success
<p>Queries all trunking ports .</p> <p>The function can be used to retrieve information about which ports are used as trunkline and which ports are enabled for trunking.</p> <p>Hint: (requires version 6.90 or later)</p>	GetTrunkPorts	<code><param><value><string>{TransKey}</string></value></param></code>	<code><param> <value><array><data> <value><string>{TransKey}</string></value> <value><array><data><value><struct> <member> <name>EnabledForTrunking</name> <value><boolean>...</boolean></value> </member> <member> <name>NetName</name> <value><string>...</string></value> </member> <member> <name>NetTrAddr</name> <value><i4>...</i4></value> </member> <member> <name>PortDispName8</name> <value><string>...</string></value> </member> <member> <name>PortTrAddr</name> <value><i4>...</i4></value> </member> <member> <name>UsedAsTrunkLine</name> <value><boolean>...</boolean></value> </member> </struct></value></data></array></value> </data></array></value></param></code>

Description	Remote Procedure	Parameters	Return Value on success
<p>Queries setup information about Trunklines exist between the Riedel rings.</p> <p>Hint: (requires version 6.90 or later)</p>	GetTrunklineSetup	<code><param><value><string>{TransKey}</string></value></param></code>	<code><param> <value><array><data> <value><string>{TransKey}</string></value> <value><i4>{ErrorCode}</i4></value> <value><array><data><value><struct> <member> <name>DestNetName</name> <value><string>...</string></value> </member> <member> <name>DestNetTrAddr</name> <value><i4>...</i4></value> </member> <member> <name>DestPortName</name> <value><string>...</string></value> </member> <member> <name>DestTrAddr</name> <value><i4>...</i4></value> </member> <member> <name>SrcNetName</name> <value><string>,,,</string></value> </member> <member> <name>SrcNetTrAddr</name> <value><i4>...</i4></value> </member> <member> <name>SrcPortName</name> <value><string>...</string></value> </member> <member> <name>SrcPortTrAddr</name></code>

Description	Remote Procedure	Parameters	Return Value on success
			<pre> <value><i4>...</i4></value> </member> </struct></value></data></array></value> </data></array></value></param> </pre>
<p>Queries activity information of trunklines.</p> <p>The function can be used to retrieve information about trunkline activity including the command type a trunkline is currently used for.</p> <p>Hint: (requires version 6.90 or later)</p>	GetTrunklineActivities	<pre> <param><value><string>{TransKey}</string></value></param> </pre>	<pre> <param> <value><array><data> <value><string>{TransKey}</string></value> <value><i4>{ErrorCode}</i4></value> <value><array><data> <value><struct> <member> <name>SrcTrAddr</name> <value> <struct> <member> <name>NetTrAddr</name> <value><i4>...</i4></value> </member> <member> <name>ObjTrAddr</name> <value><i4>...</i4></value> </member> </struct> </value> </member> <member> <name>DestTrAddr</name> <value> <struct> <member> <name>NetTrAddr</name> <value><i4>...</i4></value> </member> <member> <name>ObjTrAddr</name> <value><i4></i4></value> </pre>

Description	Remote Procedure	Parameters	Return Value on success
			<pre> </member> </struct> </value> </member> <member> <name>TrunkCmdType</name> <value>{TrunkCmdType}</value> </member> <member> <name>TrunklinePath</name> <value> <array> <data> <value> <struct> <member> <name>sNetTrAddr</name> <value><i4>...</i4></value> </member> <member> <name>sPortTrAddr</name> <value><i4>...</i4></value> </member> <member> <name>dNetTrAddr</name> <value><i4>...</i4></value> </member> <member> <name>dPortTrAddr</name> <value><i4>...</i4></value> </member> </struct> </value> </data> </array> </value></member></struct> </pre>

Description	Remote Procedure	Parameters	Return Value on success
			<pre></value></data></array></value> </param></pre>
<p>Queries all trunking ifbs .</p> <p>The function can be used to retrieve information about trunking ifbs.</p> <p>Hint: (requires version 6.90.RR5 or later)</p>	GetTrunkIfbs	<pre><param><value><string>{TransKey}</string></value></param></pre>	<pre><param> <value><array><data> <value><string>{TransKey}</string></value> <value><array><data><value><struct> <member> <name>NetTrAddr</name> <value><int>...</int></value> </member> <member> <name>IfbNumber</name> <value><int>...</int></value> </member> <member> <name>EnabledForTrunking</name> <value><boolean>...</boolean></value> </member> <member> <name>DispName8</name> <value><string>...</string></value> </member> <member> <name>LocalNetName</name> <value><string>...</string></value> </member> <member> <name>LocalNetTrAddr</name> <value><int>...</int></value> </member> <member> <name>ObjId</name> <value><int>...</int></value> </member></pre>

Description	Remote Procedure	Parameters	Return Value on success
			<pre></struct></value></data></array></value> </data></array></value></param></pre>

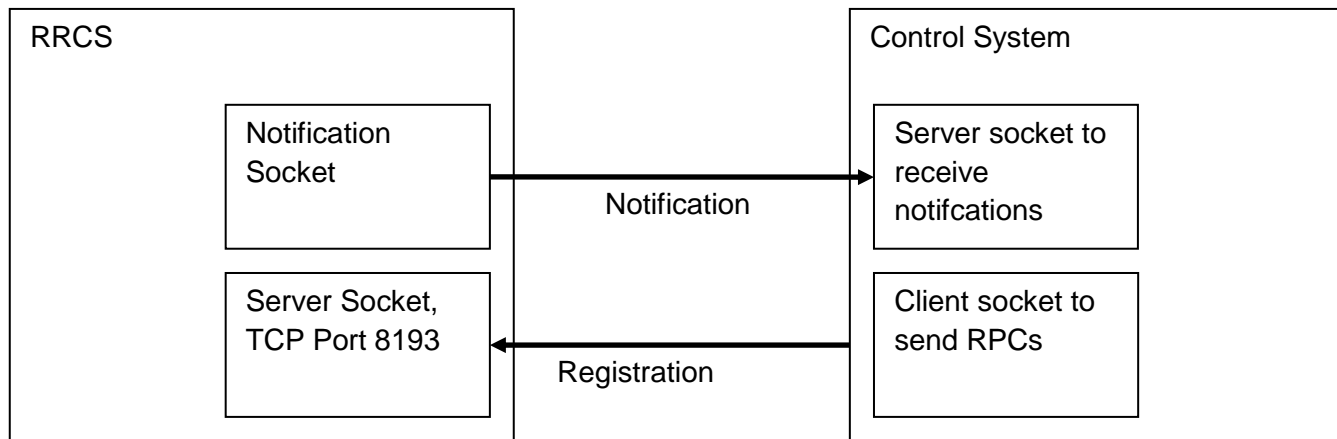
8.17 Dial / HangUp of Call

Works only if one PoolPort is configured. Configuring a number on a PoolPort can only be done via Director and is a prerequisite for these functions.

Description	Remote Procedure	Parameters	Return Value on success
<p>Instructs the system to dial the specified number.</p> <p>typically the returned LineState will be 2 (Waiting for Connection).</p> <p>Hint: (requires version 7.20 or later)</p>	DialNumber	<pre><param><value><string>{TransKey}</string></value></param> <param><value><int>{Node}</int></value></param> <param><value><int>{Port}</int></value></param> <param><value><string>Number</string></value></param></pre>	<pre><param><value><array><data> <value><string>{TransKey}</string></value> <value><int>{ErrorCode}</int></value> <value><int>{LineStatus}</int></value> </data></array></value></param></pre>
<p>Instructs the system to draw the call.</p> <p>Hint: (requires version 7.20 or later)</p>	HangUpCall	<pre><param><value><string>{TransKey}</string></value></param> <param><value><int>{Node}</int></value></param> <param><value><int>{Port}</int></value></param></pre>	<pre><param><value><array><data> <value><string>{TransKey}</string></value> <value><int>{ErrorCode}</int></value> </data></array></value></param></pre>
<p>Queries information about the status of line</p> <p>Hint: (requires version 7.20 or later)</p>	LineStatus	<pre><param><value><string>{TransKey}</string></value></param> <param><value><int>{Node}</int></value></param> <param><value><int>{Port}</int></value></param></pre>	<pre><param><value><struct> <member><name>ErrorCode</name> <value><int>{ ErrorCode }</int></value> </member> <member><name>Status</name> <value><int>{LineStatus}</int></value> </member> </struct></value></param></pre>

9 NOTIFICATIONS

The notifications described in this chapter are used by RRCS in order to signal status changes of the Artist system. External control systems need to register for notifications beforehand, please see chapter 8.15. After registration RRCS will send XML-RPCs to the given IP- and TCP address, so RRCS acts as a client and the external control system as a server.



9.1 Send String

Description	Remote Procedure	Parameters	Return Value on success
Dump/send string notification. It is sent when the Send String command is executed in the system.	SendString	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><string>{String}</string></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Dump/send string off notification. It is sent when the Send String command is executed in the system.	SendStringOff	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><string>{String}</string></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>

9.2 GPIOs

Description	Remote Procedure	Parameters	Return Value on success
Notification for a GP input change.	GpInputChange	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code> <code><param><value><int>{Slot}</int></value></param></code> <code><param><value><int>{GPIO no.}</int></value></param></code> <code><param><value><boolean>{state}</boolean></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Notification for a GP output change.	GpOutputChange	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>

Description	Remote Procedure	Parameters	Return Value on success
		<code><param><value><int>{Slot}</int></value></param></code> <code><param><value><int>{GPIO no.}</int></value></param></code> <code><param><value><boolean>{state}</boolean></value></param></code>	

9.3 Logic Source

Description	Remote Procedure	Parameters	Return Value on success
Notification for a Logic Source state change.	LogicSourceChange	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Object ID}</int></value></param></code> <code><param><value><boolean>{state}</boolean></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>

9.4 Volume change

Description	Remote Procedure	Parameters	Return Value on success
Notification for crosspoint volume changes	XpVolumeChange	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><array><data></code> <code><!-- 1st volume change information --></code> <code><value><struct></code> <code><member></code> <code><name>Source</name></code> <code><value>{TPortAddress}</value></code> <code></member></code> <code><member></code> <code><name>Destination</name></code> <code><value>{TPortAddress}</value></code> <code></member></code> <code><!-- this member only exists, if the single volume has</code> <code>changed. If this member does not exist, the member</code>	No return value (ignored by RRCS)

Description	Remote Procedure	Parameters	Return Value on success
		<pre> ConferenceVolume must exist. --> <member> <name>SingleVolume</name> <!-- -1: volume unavailable 0: volume mute 1.255: volume [dB] = (x - 230.0) / 2.0 --> <value>{x}</value> </member> <!-- this member only exists, if the conference volume has changed. If this member does not exist, the member SingleVolume must exist. --> <member> <name>ConferenceVolume</name> <!-- -1: volume unavailable 0: volume mute 1.255: volume [dB] = (x - 230.0) / 2.0 --> <value>{x}</value> </member> </struct></value> <!-- further volume change information --> ... </data></array></value> </param> </pre>	

9.5 Configuration change

Description	Remote Procedure	Parameters	Return Value on success
Notification for a configuration change.	ConfigurationChange	<param><value><string>{TransKey}</string></value></param>	No return value (ignored by RRCS)

9.6 Crosspoint change

Description	Remote Procedure	Parameters	Return Value on success
Notification for a crosspoint change.	CrosspointChange	<param><value><string>{TransKey}</string></value></param> <param><value><int>{Number of XPs}</int></value></param> <param><value><struct> <member><name>XP#1</name> <value><array><data> <value><int>{Source Net}</int></value> <value><int>{Source Node}</int></value> <value><int>{Source Port}</int></value> <value><int>{Dest Net}</int></value> <value><int>{Dest Node}</int></value> <value><int>{Dest Port}</int></value> <value><boolean>{State}</boolean></value> </data></array></value> </member> <member><name>XP#2</name> . . . </struct></value></param>	<param><value><array><data> <value><string>{TransKey}</string></value> <value><int>{ErrorCode}</int></value> </data></array></value></param>

9.7 Alarms

Description	Remote Procedure	Parameters	Return Value on success
Upstream failed on Node	UpstreamFailed	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Upstream failed cleared	UpstreamFailedCleared	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Downstream failed on Node	DownstreamFailed	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Downstream failed on Node	DownstreamFailedCleared	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Node Controller failed on Node	NodeControllerFailed	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Node Controller reboot on Node	NodeControllerReboot	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Client failed on Node	ClientFailed	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code>

Description	Remote Procedure	Parameters	Return Value on success
		<code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Slot-of-client-card}</int></value></param></code>	<code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Client failed on Node cleared	ClientFailedCleared	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Slot-of-client-card}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Port went inactive	PortInactive	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Port went active	PortActive	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><int>{Net}</int></value></param></code> <code><param><value><int>{Node}</int></value></param></code> <code><param><value><int>{Port}</int></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Connect to Artist net succesful after failure	ConnectArtistRestored	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><string>{GatewayState}</string></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Connect to Artist net failure	ConnectArtistFailure	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><string>{GatewayState}</string></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>
Gateway shutdown due to application closure or PC shutdown.	GatewayShutdown	<code><param><value><string>{TransKey}</string></value></param></code>	<code><param><value><array><data></code> <code><value><string>{TransKey}</string></value></code> <code><value><int>{ErrorCode}</int></value></code> <code></data></array></value></param></code>

9.8 *GetAlive*

RRCS calls this method periodically if there was no need to send a notification during the last second. If RRCS does not receive a response it removes the notification receiver from its internal receiver list. In this case a call from an external control system to RRCS of the method 'IsRegisteredForAllEvents' will return false.

Note: This notification is sent only if 'RegisterForAllEvents' has been successfully called before.

Description	Remote Procedure	Parameters	Return Value
Ping	GetAlive	none	ignored by RRCS

9.9 Panel Spy

9.9.1 Overview

An external control system needs to perform the following two registration steps to be able to receive panel spy events from Artist-panels through RRCS:

- 1) The control system calls 'RegisterForAllEvents'
- 2) The control system calls 'ChangePanelSpyRegistry' for at least one panel

Once these steps have been completed successfully, RRCS informs the external control system about the notification status for each panel (see 'PanelSpyStateChange') and calls the notification handlers if necessary (see 'PanelSpyRotateEvent', 'PanelSpyKeyEvent', 'PanelSpyFuncKeyEvent' and 'PanelSpyNumKeyEvent'). RRCS watches the connection state to artist and the online state of all hardware components needed to be able to receive panel spy events and synchronizes the panel spy registration table with artist automatically. Therefore there is no need to reregister anything if for example a node-controller or a client-card reboots.

However, if RRCS cannot reach the external control system (if the external control system does not respond to the 'GetAlive' notification because, for instance, a network switch has failed) then no further notifications are sent. Therefore an external control system should periodically ask RRCS via 'IsRegisteredForAllEvents' whether the notification channel is still active or not and should restart the two registration steps on deactivation.

RRCS places no limit on the number of external control systems or on the number of panel spy receivers for each control system. If a control system registers itself multiple times for a single panel, it will receive multiple copies of the same panel notifications.

9.9.2 PanelSpyStateChange

Description	Remote Procedure	Parameters	Return Value
Notifies panel spy status changes for a list of panels	PanelSpyStateChange	<pre> <param><value><string>{TransKey}</string></value></param> <param><value><array><data> { panel spy state of 1st panel (see below)} { panel spy state of 2nd panel (see below)} ... </data></array></value></param> </pre>	ignored by RRCS

A single panel spy state is a xml-rpc-struct which contains the following members:

Name	Type	Parameter description
Node	Int	The panel's node address
Port	int	The panel's port address
Rotate	struct	Defines rotate event type status (see below)
Key	struct	Defines key event type status (see below)
FuncKey	struct	Defines function key event type status (see below)
NumKey	struct	Defines the numerical key event type status (see below)

The rotate/key/function key and numerical key status is defined as follows (xml-rpc-struct):

Name	Type	Parameter description
State	int	<ul style="list-style-type: none"> 0=unregistered: Panel spy not registered 1=busy: Panel spy registered: RRCS tries to activate the panel spy in artist. This is always the first notification-state after a successful ChangePanelSpyRegistry call 2=active: The control system receives panel spy notifications

		<ul style="list-style-type: none"> 3=error: The control system does not receive panel spy notifications (however, RRCS still tries to activate the panel spy). else: undefined value
ErrorCode	int	Defines the error code. This struct member exists only if the state member indicates an error (=3)
ErrorDescription	string	A human readable error reason. This struct member exists only if the state member indicates an error (=3)

9.9.3 PanelSpyRotateEvent

Description	Remote Procedure	Parameters	Return Value
Rotary encoder change notification	PanelSpyRotateEvent	<param><value><string>{TransKey}</string></value></param> <param><value><struct>{RotateEventData}</struct></value></param>	ignored by RRCS

RotateEventData:

Name	Type	Parameter description																		
Node	int	The panel's node address																		
Port	int	The panel's port address																		
SubPanel	int	Specifies whether the event occurred on the main panel (SubPanel = 0) or from an expansion panel. If the event occurred on an expansion panel a value of - 1 means: 1st expansion panel - 2 means: 2nd expansion panel - ... - 6 means: 6-th expansion panel Other values are invalid																		
Key	int	Defines key number (< 0 means: rotary master)																		
RotationTicks	int	The number of rotation ticks (positive values == rotate right)																		
FuncKeyStates	struct	<table> <tr> <th>Name</th><th>Type</th><th>Parameter description</th></tr> <tr> <td>Shift</td><td>boolean</td><td>Shift page function key (key-no=0)</td></tr> <tr> <td>Hs</td><td>boolean</td><td>Headset function key (key-no=1)</td></tr> <tr> <td>Opt</td><td>boolean</td><td>Options function key (key-no=2)</td></tr> <tr> <td>Beep</td><td>boolean</td><td>Beep/F1 function key (key-no=3)</td></tr> <tr> <td>Norm</td><td>boolean</td><td>Norm/F2 function key (key-no=4)</td></tr> </table>	Name	Type	Parameter description	Shift	boolean	Shift page function key (key-no=0)	Hs	boolean	Headset function key (key-no=1)	Opt	boolean	Options function key (key-no=2)	Beep	boolean	Beep/F1 function key (key-no=3)	Norm	boolean	Norm/F2 function key (key-no=4)
Name	Type	Parameter description																		
Shift	boolean	Shift page function key (key-no=0)																		
Hs	boolean	Headset function key (key-no=1)																		
Opt	boolean	Options function key (key-no=2)																		
Beep	boolean	Beep/F1 function key (key-no=3)																		
Norm	boolean	Norm/F2 function key (key-no=4)																		

9.9.4 PanelSpyKeyEvent

Description	Remote Procedure	Parameters	Return Value
Key event notification	PanelSpyKeyEvent	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><struct>{KeyEventData}</struct></value></param></code>	ignored by RRCS

KeyEventData:

Name	Type	Parameter description								
KeyType	int	<table><tr><td>0</td><td>Standard key event, left position</td></tr><tr><td>1</td><td>Key event on standard key, right position. Only available in 1000 and 1100 series.</td></tr><tr><td>2</td><td>Rotary or master rotary key event.</td></tr></table>	0	Standard key event, left position	1	Key event on standard key, right position. Only available in 1000 and 1100 series.	2	Rotary or master rotary key event.		
0	Standard key event, left position									
1	Key event on standard key, right position. Only available in 1000 and 1100 series.									
2	Rotary or master rotary key event.									
KeyAction	int	<table><tr><td>0</td><td>Key pressed (down)</td></tr><tr><td>1</td><td>Key released (up)</td></tr><tr><td>2</td><td>Double click event. (Currently not supported. Maybe supported in future releases)</td></tr><tr><td>3</td><td>Key long-press event. (Currently not supported. Maybe supported in future releases)</td></tr></table>	0	Key pressed (down)	1	Key released (up)	2	Double click event. (Currently not supported. Maybe supported in future releases)	3	Key long-press event. (Currently not supported. Maybe supported in future releases)
0	Key pressed (down)									
1	Key released (up)									
2	Double click event. (Currently not supported. Maybe supported in future releases)									
3	Key long-press event. (Currently not supported. Maybe supported in future releases)									
Node	int	The panel's node address								
Port	int	The panel's port address								
SubPanel	int	Specifies whether the event occurred on the main panel (SubPanel = 0) or from an expansion panel. If the event occurred on an expansion panel a value of <ul style="list-style-type: none">- 1 means: 1st expansion panel- 2 means: 2nd expansion panel- ...- 6 means: 6-th expansion panel Other values are invalid								
Key	int	Defines key number (< 0 means: rotary master)								
IsKeyLatched	boolean	Tells whether the key is in latching state or not								
FuncKeyStates	struct	<table><tr><th>Name</th><th>Type</th><th>Parameter description</th></tr></table>	Name	Type	Parameter description					
Name	Type	Parameter description								

		Shift	boolean	Shift page function key (key-no=0)
		Hs	boolean	Headset function key (key-no=1)
		Opt	boolean	Options function key (key-no=2)
		Beep	boolean	Beep/F1 function key (key-no=3)
		Norm	boolean	Norm/F2 function key (key-no=4)

9.9.5 PanelSpyFuncKeyEvent

Description	Remote Procedure	Parameters	Return Value
Function key event notification	PanelSpyFuncKeyEvent	<param><value><string>{TransKey}</string></value></param> <param><value><struct>{FuncKeyEventData}</struct></value></param>	ignored by RRCS

FuncKeyEventData:

Name	Type	Parameter description		
FuncKeyAction	Int	0	function key pressed (down)	
		1	function key released released (up)	
		2	Auto-off: The function key has been turned off automatically (e.g. via a timer)	
Node	int	The panel's node address		
Port	int	The panel's port address		
FuncKeyNo	int	0	Shift	
		1	Hs	
		2	Opt	
		3	Beep	
		4	Norm	
FuncKeyStates	struct	Name	Type	Parameter description
		Shift	boolean	Shift page function key (key-no=0)
		Hs	boolean	Headset function key (key-no=1)
		Opt	boolean	Options function key (key-no=2)
		Beep	boolean	Beep/F1 function key (key-no=3)
		Norm	boolean	Norm/F2 function key (key-no=4)

9.9.6 PanelSpyNumKeyEvent

Description	Remote Procedure	Parameters	Return Value
Numerical key event notification	PanelSpyNumKeyEvent	<code><param><value><string>{TransKey}</string></value></param></code> <code><param><value><struct>{NumKeyEventData}</struct></value></param></code>	ignored by RRCS

NumKeyEventData:

Name	Type	Parameter description																
NumKeyAction	Int	<table><tr><td>0</td><td>Numerical key pressed (down)</td></tr><tr><td>1</td><td>Numerical key released released (up)</td></tr></table>	0	Numerical key pressed (down)	1	Numerical key released released (up)												
0	Numerical key pressed (down)																	
1	Numerical key released released (up)																	
Node	int	The panel's node address																
Port	int	The panel's port address																
SubPanel	int	Specifies whether the event occurred on the main panel (SubPanel = 0) or from an expansion panel If the event occurred on an expansion panel a value of - 1 means: 1st expansion panel - 2 means: 2nd expansion panel - ... - 6 means: 6-th expansion panel Other values are invalid																
NumKeyNo	int	<table><tr><td>1</td><td>M1</td></tr><tr><td>2</td><td>M2</td></tr><tr><td>3</td><td>M3</td></tr><tr><td>4</td><td>M4</td></tr><tr><td>5</td><td>M5</td></tr><tr><td>6</td><td>M6</td></tr><tr><td>7</td><td>M7</td></tr><tr><td>8</td><td>M8</td></tr></table>	1	M1	2	M2	3	M3	4	M4	5	M5	6	M6	7	M7	8	M8
1	M1																	
2	M2																	
3	M3																	
4	M4																	
5	M5																	
6	M6																	
7	M7																	
8	M8																	

		35	'#'
		42	'*'
		48	'0'
		49	'1'
		50	'2'
		51	'3'
		52	'4'
		53	'5'
		54	'6'
		55	'7'
		56	'8'
		57	'9'
		67	clear = 'C' character
		80	'P'
		82	Redial = 'R' character
		83	Speaker = 'S' character

FuncKeyStates	struct	Name	Type	Parameter description
		Shift	boolean	Shift page function key (key-no=0)
		Hs	boolean	Headset function key (key-no=1)
		Opt	boolean	Options function key (key-no=2)
		Beep	boolean	Beep/F1 function key (key-no=3)
		Norm	boolean	Norm/F2 function key (key-no=4)

10 REDUNDANCY

RRCS is designed to run in a redundant environment. The Gateway software has two states “Working” and “Standby”. The state is checked and controlled by an external Control System. To achieve that, RRCS provides a set of commands.. These “Redundancy” commands are supported on both (Working and Standby) gateways at anytime.

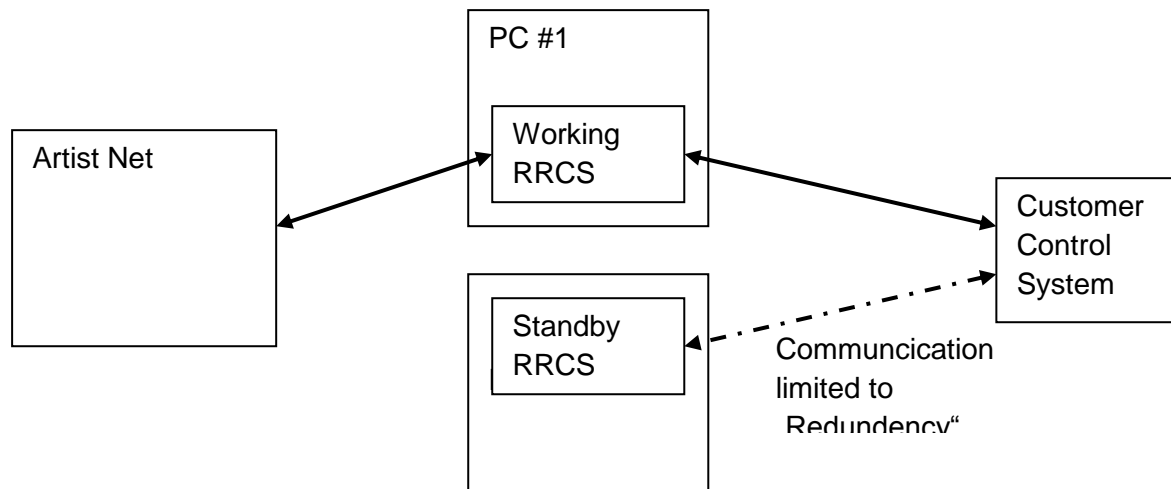
“GetState” to query the Gateway state (Working/Standby).

“SetStateWorking” and “SetStateStandby” to switchover the Gateway.

“GetAlive” to check that the gateway is still up and running.

“GetArtistConnectionState” to query the network connection state between the gateway and the Artist net.

“ConnectArtistFailure” and “ConnectArtistFailureRestored” to report the Artist connection state.



Possible scenarios that require a switchover:

- The Working Gateway PC has a power failure and does not to answer to “GetAlive” requests.
- The network between the Customer Control System and the Working network PC fails and the gateway does not to answer to “GetAlive” requests.
- The Working Gateway cannot connect to Artist net and reports “ConnectArtistFailure”.

A switchover to the the Standby Gateway only makes sense, if it’s in a good condition. That means the Standby Gateway answers to “GetAlive” requests and the network connection to Artist is ok. If the Control System initiates a switchover the Working gateway should be set to Standby (if possible), before the Standby gateway is set to Working. The case that there are two Working gateways at the same time must be avoided, because this could cause undefined results in the Artist network.

11 COMMUNICATION EXAMPLES

11.1 *Example for an incoming request to RRCS*

This is an example for setting a Crosspoint request between source port 35 on node 2 and destination port 69 on node 5. Only the XML data is shown.

```
<?xml version="1.0"?>

<methodCall>

  <methodName>SetXp</methodName>

  <params>

    <param><value><string>C0002817191</string></value></param>

    <param><value><int>1</int></value></param>

    <param><value><int>2</int></value></param>

    <param><value><int>19</int></value></param>

    <param><value><int>1</int></value></param>

    <param><value><int>5</int></value></param>

    <param><value><int>3</int></value></param>

  </params>

</methodCall>
```

Response:

```
<?xml version="1.0"?>

<methodResponse>

  <params>

    <param>

      <value><array><data>
```

```
<value><string>C0002817191</string></value>

<value><int>0</int></value>

</data></array></value>

</param>

</params>

</methodResponse>
```

11.2 Example for an outgoing request from RRCS

This is an example for an “Upstream Failed” event.

```
<?xml version="1.0"?>

<methodCall>

  <methodName>UpstreamFailed</methodName>

  <params>

    <param>

      <value><string>R1947584733</string></value>

    </param>

    <param>

      <value><int>2</int></value></param>

    </params>

  </methodCall>
```

Response:

```
<?xml version="1.0"?>
```

```
<methodResponse>  
  
  <params>  
  
    <param>  
  
      <value><array><data>  
  
        <value><string>R1947584733</string></value>  
  
        <value><int>0</int></value>  
  
      </data></array></value>  
  
    </param>  
  
  </params>  
  
</methodResponse>
```

12 PERFORMANCE

The time to perform an individual remote procedure call is related to several variables. The remote procedure calls can be split into four categories.

12.1 Querying information

The performance of RPCs querying system information is related to the network and PC processing performance. If the PC and network are idle, these RPCs are typically answered within 10ms.

These RPCs are querying information:

GetXpStatus

GetAllActiveXps

GetActiveXpsRange

GetXpVolume

GetPortAlias

GetInputGain

GetOutputGain

GetGpInputState

GetGpOutputState

GetAllCaps

GetAllLogicSources

GetObjectList

12.2 *Pure RRCS communication*

Some RPCs are purely related to RRCS internal states and are related to the network and PC processing performance. If the PC and network are idle, these RPCs are typically answered within 1ms.

GetState

GetVersion

SetStateWorking

SetStateStandby

RegisterForEvents

UnregisterForEvents

RegisterForGpInputChange

UnregisterForGpInputChange

RegisterForGpOutputChange

UnregisterForGpOutputChange

RegisterForEventsEx

UnregisterForEventsEx

12.3 *Changing system states*

The performance of RPCs querying system information is related to the network performance, PC processing performance and usage of the Artist system. If the PC, network and Artist are idle, these RPCs are typically answered within 50ms.

These RPCs are changing the system state:

SetXp

SetXpPrio

SetXpDestructive

KillXp

SetXpVolume

SetInputGain

SetOutputGain

SetGpOutput

SetLogicSourceState

12.4 *Changing system configuration*

The performance of RPCs querying system information is related to the network performance, PC processing performance, the system configuration size and the usage of the Artist system. If the PC, network and Artist are idle, these RPCs are typically answered within several seconds. The response time will be much longer, if other PCs change the configuration at the same time.

These RPCs are changing the system configuration:

SetPortAlias

ConfigurationChange