



.: www.totalrecallvr.com :.

Audit REST API

User Guide

March, 2021

Guide Issue 7.0

Total Recall VR Release 11.11.0

Copyright © 2021 Prolancer Pty Ltd, Sydney, Australia.

The text of and illustrations in this document are licensed by Prolancer Pty Ltd under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at <http://creativecommons.org/licenses/by-sa/3.0/>. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version. Prolancer, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Total Recall VR™ is a trade mark of Prolancer Pty Ltd. in Australia and other countries. All other trademarks are the property of their respective owners.

The information in this publication is subject to change without notice. Prolancer Pty Ltd assumes no responsibility for any errors that may appear in this publication.



Prolancer

Related Documents:

[1] Prolancer Pty Ltd, Total Recall VR web site. Available from:
<http://www.totalrecallvr.com/>.

[2] Prolancer Pty Ltd, Prolancer web site. Available from:
<http://www.prolancer.com.au/>.

[3] Prolancer Pty Ltd, Total Recall VR Overview User Guide, 27.0, March 2021

[4] Prolancer Pty Ltd, Total Recall VR Manager User Guide, 18.0, March 2021

Table of Contents

1. Preface	4
1.1. <i>Conventions</i>	4
1.1.1. Notes & Warnings	4
1.1.2. Typographic Conventions.....	4
1.1.3. Procedures	5
1.2. <i>We Need Feedback</i>	5
2. Introduction	6
2.1. <i>About This Guide</i>	6
2.2. <i>What is Total Recall VR</i>	6
2.3. <i>What is the Audit REST API?</i>	8
3. Start Here	9
3.1. <i>URI Structure</i>	9
3.2. <i>Resource Representation</i>	9
3.3. <i>API Documentation</i>	11
3.4. <i>Authentication</i>	11
3.5. <i>Paginations</i>	11
3.6. <i>Ordering</i>	12
3.7. <i>Filters</i>	12
3.8. <i>Resource Identification</i>	12
3.9. <i>Resource Versioning</i>	12
4. Resources	13
4.1. <i>Audit Events</i>	13
4.1.1. Schema	13
4.1.2. GET /rest/api/1/events	13
4.1.3. GET /rest/api/1/events/{id}	14

1. Preface

1.1. Conventions

Our guides use several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

1.1.1. Notes & Warnings

We use the following visual styles to draw attention to information that might otherwise be overlooked:



Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.



Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring the information will not cause data loss, but may cause irritation and frustration.



Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

1.1.2. Typographic Conventions

We use typographic conventions to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

Example	Meaning
Select <u>Guide</u> to display ...	Locate the link named "Guide" on the screen, position the cursor over the link and then depress the appropriate mouse button to follow the link.
Select Add to create a new ...	Locate the button or menu item named "Add" on the screen, position the cursor over the button or menu item and then depress the appropriate mouse button to initiate an action.

Enter <i>Commission</i> ...	Locate the field named "Commission" on the screen, position the cursor over the field and then depress the appropriate mouse button to select the field. Once the cursor appears in the field, enter a value.
Choose <i>Country</i> ...	Locate the field named "Country" on the screen, position the cursor over the field and then depress the appropriate mouse button to display the available options. Then position the cursor over the desired option and depress the appropriate mouse button to select it.
Tick <i>Active User</i> ...	Locate the check box named "Active User" on the screen, position the cursor over the check box and depress the appropriate mouse button to place a visual tick in the box.
Un-tick <i>Active User</i> ...	Locate the check box named "Active User" on the screen, position the cursor over the check box and depress the appropriate mouse button to remove the visual tick in the box.
Enter \$30.95 ...	Enter "\$30.95" using the keys on your keyboard.

1.1.3. Procedures

We use numbered sequence of steps to define procedures for performing certain tasks. For example:

Procedure Title

1. This is the first step of the procedure.
2. This is the second step of the procedure.
 - a. This is the first sub-step of step 2.
 - b. This is the second sub-step of step 2.
3. This is step three.

1.2. We Need Feedback

If you find a typographical error in this guide, or if you have thought of a way to make this guide better, we would love to hear from you.

Please submit your feedback using the feedback form on our web site:

<http://www.prolancer.com.au/contact/feedback>.

If you have a suggestion for improving the guide, then try to be as specific as possible when describing your suggestion. Otherwise, if you have found an error, please include the section number and some of the surrounding text so we can find it easily.

2. Introduction

2.1. About This Guide

This guide describes the Total Recall VR Audit REST API in a form of an API reference.

The guide is intended for integrators and developers who are integrating Total Recall VR recorders with other business systems, or script interactions with Total Recall VR recorders for the purpose of auditing and reporting.

2.2. What is Total Recall VR



Audio Logging
and
Call Recording
Systems

Total Recall VR is a professional audio logging and call recording system which is self-contained, fully featured and cost-effective. Enterprises and governments worldwide use it to create electronic records of many forms of audio communication including telephone, 2-way radio, broadcast radio, public address, intercoms, room microphones and much more.

Total Recall VR is the ideal solution for:

- Recording business telephone conversations;
- Recording agent calls in contact centres;
- Logging emergency response communication;
- Logging business operations communication;
- Logging radio broadcasts;
- Logging public announcements;
- Logging Air Traffic Control communication;
- Creating audio records of meetings, legal proceedings, public enquiries and similar events; and
- Creating compliance records to meet duty of care and legal requirements.

Total Recall VR captures all audio in digital format and stores it in a proprietary, secure and tamper proof file format in its on-board hard drive storage. The file format preserves the originality of the audio that it stores and has a number of built-in mechanisms that

aid quick and reliable detection of tampering. However, for ease of access, Total Recall VR client applications can generate copies of recordings in a number of popular and everyday formats such as Microsoft's Wave (.wav) and MPEG Layer-3 (.mp3).

Storing audio by itself does not help when looking for one recording in a store that can hold hundreds of thousands of recordings. That is why, in addition to audio, Total Recall VR captures and then stores information related to each recording and audio source in its database such as start time, end time and duration of recordings, calling and called numbers on telephone calls, DTMF digits during calls, radio IDs, user configurable notes and much more. This information is the backbone of a powerful search capability which can pin point a single recording in a set of hundreds of thousands of recordings which reside either on a Total Recall VR recorder or in one of many types of off-system archives of recordings.

In addition to the audio recorder and the on-board storage, each Total Recall VR system comes with a built-in media player with comprehensive player controls (start, stop, fast-forward, rewind ...). The player can play audio stored in files directly on the system or stream audio to a remote client application which then outputs the sound to the PC speakers of the PC that it runs on.

While audio recording, storage and re-play are the main functions of Total Recall VR, every Total Recall VR offers many more advanced, professional-grade features. For example:

- Ability to capture audio from different types of audio sources (analogue, VoIP, RoIP, AoIP and ISDN), at the same time – hybrid recording.
- Live and real-time monitoring (listening) of recordings in progress on the system itself or on a remote PC with the aid of a PC client application.
- Feature-rich archiver which can create searchable archives of recordings on CD, DVD or BD discs, USB keys or drives and network drives, either automatically or on-demand.
- Automated self-cleaning mechanism that removes obsolete recordings automatically and on regular intervals to keep the system operating endlessly.
- SNMP agent capable of generating SNMP alarms (traps).
- SMDR integration for a number of popular PBXes.
- Fully internationalised user interface; all menus and software available in multiple languages.
- Role based access control.
- On-board LCD display and control keypad on selected models.
- A range of PC client applications with value adding functionality.
- APIs and PC applications for integration with other business systems and solutions.

When audio records are critical to your operations, Total Recall VR delivers. It is professional, reliable and fully self-contained solution for audio logging and call recording that comes at an affordable price.



The Total Recall VR Overview [3] guide contains a comprehensive description and overview of Total Recall VR.

2.3. What is the Audit REST API?

The Total Recall VR Audit REST API is an API which provides access to system and user events that are stored on Total Recall VR recorders.

The events describe “who did what and when”. As a result, the information captured by the events can be used to conduct systems audits and for audit reporting in general.

This guide documents the audit REST resources on Total Recall VR recorders along with available HTTP operations on the resources and expected responses.

3. Start Here

3.1. URI Structure

The API provides access to resources via URI paths. To use the API, your application will make an HTTP request and parse the response that it receives.

URIs for Total Recall VR resources have the following structure:

```
http://<host>:<port>/<context>/rest/api/<version>/<resource>
```

Where:

- <host> - the IP address assigned to either the LAN 1 or the LAN 2 interface of the recorder.
- <port> - 4020.
- <context> - the context for the Audit API is 'audit'.
- <version> - the current version of the Audit API is 1. However, there is also a symbolic version called 'latest' which resolves to this API version.
- <resource> - a REST resource that is accessible via the API. Subsequent sections of this document describes the resources that are accessible via the Audit API.



The Total Recall VR Manager User Guide [4] contains information on how to configure the LAN 1 and LAN 2 interface on Total Recall VR recorders.

For example:

```
http://192.168.130.202:4020/audit/rest/api/1/events
```

```
http://192.168.130.202:4020/audit/rest/api/latest/events
```

```
http://trvr-202.tsn.prolancer.com.au:4020/audit/rest/api/1/events
```

```
http://trvr-202.tsn.prolancer.com.au:4020/audit/rest/api/latest/events
```

3.2. Resource Representation

The API can provide resources in a number of representations. Your application must set the Accept header to one of the supported representation formats. Supported formats:

Format	Example
application/xml or application/*	<pre> <auditEvent> <id>1063</id> <version>1</version> <eventCategory>WARN</eventCategory> <eventPriority>HIGH</eventPriority> <eventDateTime>20180406T072917.065+1000</eventDateTime> <eventSource>recorder 201</eventSource> <eventActor>admin</eventActor> <eventAction>detach</eventAction> <actionSubject>disk</actionSubject> <actionDataSet> <actionData> <id>1884</id> <version>1</version> <tag>disk</tag> <value>/dev/sdb</value> </actionData> </actionDataSet> </auditEvent> </pre>
application/json	<pre> { "id": 1063, "eventSource": "recorder 201", "eventAction": "detach", "eventPriority": "HIGH", "version": 1, "actionSubject": "disk", "eventDateTime": "20180405T214815.990+0000", "eventActor": "admin", "eventCategory": "WARN", "actionDataSet": [{ "tag": "disk", "version": 1, "value": "/dev/sdb", "id": 1884 }] } </pre>
application/hal+json	<pre> { "_links": {"self": {"href": "/rest/api/1/events/1063"}}, "id": 1063, "eventSource": "recorder 201", "eventAction": "detach", "_embedded": {"actionDataSet": [{ "tag": "disk", "version": 1, "value": "/dev/sdb", "id": 1884 }]}, "eventPriority": "HIGH", "version": 1, "actionSubject": "disk", "eventDateTime": "20180405T214815.990+0000", "eventActor": "admin", "eventCategory": "WARN" } </pre>

3.3. API Documentation

Swagger documents for the API are available via the following URLs:

`http://<host>:<port>/audit/swagger.json`

`http://<host>:<port>/audit/swagger.yaml`

3.4. Authentication

None supported at this stage.

3.5. Paginations

The API enforces pagination for all methods that can return a large collection of resources. Methods that enforce pagination return at most one page of data for each request.

Pagination can be controlled by the following query parameters:

- `pageNo` – the actual page of resources to return. Default value is 1 (the 1st page).
- `pageSize` – the maximum number of resources to include on the page. Default value is 100.

Clients that request the `application/xml` or the `application/json` representation of resources, need to store the page number and size of the last request in their own state and use those values to calculate the page number and size for subsequent requests.

Clients that request the `application/hal+json` representation will receive page metadata as part of the response, for example:

```
{
  "pageNumber": 5,
  "itemsPerPage": 10,
  "pagesAvailable": 11,
  "itemsAvailable": 106,
  "_embedded": {"auditEvents": [
    { ... }
    { ... }
    { ... }
  ]},
  "_links": {
    "self": {"href": "/rest/api/1/events?pageNo=5&pageSize=10"},
    "next": {"href": "/rest/api/1/events?pageNo=6&pageSize=10"},
    "previous": {"href": "/rest/api/1/events?pageNo=4&pageSize=10"},
    "last": {"href": "/rest/api/1/events?pageNo=11&pageSize=10"},
    "first": {"href": "/rest/api/1/events?pageNo=1&pageSize=10"}
  }
}
```

Where:

- `pageNumber` – the number of the page from the request.

- `itemsPerPage` – the number of items per page from the request.
- `pagesAvailable` – the number of available pages with the requested page size at the time of the request.
- `itemsAvailable` – the total number of resources at the time of the request.

In addition, as the example shows, the response will contain a number of links which makes it easier to move between pages.

3.6. Ordering

Resources are always ordered in descending time of event order, that is, most recent event first.

3.7. Filters

The API supports Feed Item Query Language (FIQL) filters which can be specified with the filter query parameter when making requests. For example:

```
http://.../audit/rest/api/1/events?filter=eventCategory==WARN
```

3.8. Resource Identification

Resources of all types have an “id” property which is a unique integer value across all resources of the same type.

3.9. Resource Versioning

Resources of all types have the “version” property. The integer value of this property for a resource is incremented every time the resource is modified. As a result, the value of the property can be used to detect when a stale resource representation is in use.

4. Resources

The information in this section is introductory only. Please see the Swagger documentation for full details including schemas.



To access the Swagger documentation use:

`http://<host>:<port>/audit/swagger.json`

`http://<host>:<port>/audit/swagger.yaml`

4.1. Audit Events

4.1.1. Schema

See the Swagger documents for the API which are available via the following URLs:

`http://<host>:<port>/audit/swagger.json`

`http://<host>:<port>/audit/swagger.yaml`

4.1.2. GET /rest/api/1/events

Returns a page (collection with specified size) of events in one of the supported representation formats.

Path parameters:

None.

Query parameters:

Parameter	Type	Description
pageNo	integer	The page to fetch. Valid values are positive integers starting with 1. The default value is 1.
pageSize	integer	The maximum number of resources to include in the response. Valid values are positive integers starting with 1. The default value is 100.
filter	FIQL expression	The filter that resources must match in order to be included in the response.

Responses:

Response	Description
200	The response contains a page of resources in the specified representation. Note that the response may contain less than pageSize resources.
404	The requested page is an empty page.
500	The request could not be processed due to an internal problem.

4.1.3. GET /rest/api/1/events/{id}

Returns the audit event which has an “id” parameter that matches the {id} part of the request.

Path parameters:

Parameter	Type	Description
id	integer	Value of the “id” parameter of an audit event.

Request parameters:

None.

Responses:

Response	Description
200	The response contains the specified resource in the specified representation.
404	The requested resource does not exist.
500	The request could not be processed due to an internal problem.

[End of Document]