Capstone Project for Microsoft Graph Call Records API

This document serves as a capstone project to practice and enhance technical writing skills. The goal is to submit it for review and obtain a Technical Writer certificate from Technical Writer HQ. The capstone project includes seven different types of documentation and covers various aspects of the Microsoft Graph Call Records API.

Table of Contents

USER PERSONA: ANNIE CHEN, TEAMS ADMINISTRATOR WITH MICROSOFT GRAPH EXPERTISE

Name: Annie Chen

Age: 36

Education: Master's degree in computer science

Role: Microsoft Teams Administrator

Location: Seattle, WA

Annie Chen, an experienced **Microsoft Teams administrator**, works in the IT department of a **medium-sized organization**. With **7 years of experience**, she holds **certifications** in **Microsoft 365** and **Teams administration**. Enthusiastic about optimizing collaboration and communication, Annie stays abreast of the latest features and capabilities of Microsoft Teams and Graph APIs. As an expert user, she **develops** third-party app based on the Graph Call Records API, **troubleshoots** call quality issues, and provides **support** and **education** to users.

In addition, Annie actively contributes **to attracting women to the IT field** by participating in initiatives that promote **diversity and inclusion**. She is also a lecturer on communication and development topics at IT conferences.

RESPONSIBILITIES:

- Monitors call quality metrics (audio/video clarity, dropped calls, connectivity)
- Analyzes call records data for trends affecting call quality
- Collaborates with network engineers to troubleshoot issues
- Utilizes the Graph Call Records API for detailed call information
- Creates custom reports and dashboards
- Conducts quality assurance tests on Teams calls
- Educates end-users on call quality best practices

GOALS & NEEDS:

- Improve overall call quality by identifying and addressing performance bottlenecks to enhance call clarity, reliability, and user experience
- Educate end-users on best practices to maximize call quality
- Enhance end-user satisfaction through seamless communication experiences

CHALLENGES:

- Balancing call quality improvements with budget constraints
- Troubleshooting and interpreting call data related to complex call quality scenarios
- Addressing call quality issues reported by end-users without full visibility into their experiences

SKILLS & TOOLS:

- **Administration**: Microsoft 365 and Teams
- **Microsoft Graph**: Comfortable with Graph Explorer, SDKs, and REST APIs
- **Programming Languages**: C#, Python, PowerShell (for 3rd-party app development and Teams administration tasks)
- **Azure**: Utilizes Azure Active Directory for user management and permissions
- **Dashboard Tools**: Power BI for data visualization

MOTIVATION:

- **Driving Organizational Impact**: Annie is motivated by the impact of effective collaboration on organizational productivity and employee satisfaction
- **Thriving on Challenges**: Annie enjoys solving real-world challenges using Microsoft Teams and Graph capabilities
- **Empowering Diversity**: Annie's passion extends beyond technology; she actively contributes to attracting women to the IT industry through conferences and initiatives

KNOWLEDGE BASE: MICROSOFT GRAPH CALL RECORDS API

Troubleshoot call quality with Graph Call Records API

Microsoft Graph Call Records API offers usage and diagnostic insights for calls made within a business organization using Microsoft Teams. Find information and guidance about Graph Call Records API.

Search...

Key Features

Retrieve detailed call records, subscribe to notifications for new call records, and use the API to gather diagnostic data for troubleshooting and monitoring call quality.

APIs

- Get callRecord
- List callRecords
- List participants_v2

Use Cases

- Peer-to-peer scenario interpretation
- Build participants list
- High-Level Architecture

FAQ

Find answers to common questions about using the Graph Call Records API.

RESOURCES

Training and Tutorials

- How to call the API
- How to authenticate
- How to add permissions
- How to interpret a scenario

Support

- Microsoft Graph support
- Microsoft Graph Q&A community support
- Microsoft Graph feedback portal
- Microsoft Graph Known issues
- Microsoft Graph Tech Community
- Microsoft 365 Developer Blog

API: MICROSOFT GRAPH CALL RECORDS API

Microsoft Graph Call Records API offers usage and diagnostic insights for calls made within a business organization that uses Microsoft Teams or Skype for Business. A call record provides detailed information about the call, such as start and end time, joined participants, devices, network, and other aspects. Additionally, the call record reflects user feedback on call quality and captures quality metrics for audio, video, and screen sharing during the call. The Call Records API offers many more details that can be used by administrators to assess overall call quality and troubleshoot issues.

Resource

GET */communications/callRecords/**{id}***

Get call diagnostic information by call id

Parameters

Path parameters

Parameter	*Required/Optional*	*Description*
id	Required	Id of the call made in the organization

Query string parameters

Parameter	*Required/Optional*	*Description*
$expand	Optional	Use the **$expand** query parameter to include the expanded resource of participants_v2, sessions or segments.
$select	Optional	Use the **$select** query parameter to return a set of properties, for example, to overview id and startDateTime. Only supported for callRecord and session resources.

Request Header parameters

Parameter	*Required/Optional*	*Description*
Authorization	Required	Bearer token. How to get the token instruction.
Prefer: odata.maxpagesize={x}	Optional	Specifies a preferred integer {x} page size for paginated results. This value must be equal to or less than 60.
Prefer: omit-values=nulls	Optional	Removes null or empty values from the call record response.

Example Request

GET *https://graph.microsoft.com/v1.0/communications/callRecords/e523d2ed-2966-4b6b-925b-754a88034cc5*

(**Note**: replace *e523d2ed-2966-4b6b-925b-754a88034cc5* with your actual call id)

Example Response

This is a sample response from the */communications/callRecords/**{id}*** endpoint:

```
HTTP/1.1 200 OK
Content-type: application/json
{
  "@odata.context":
"https://graph.microsoft.com/v1.0/$metadata#communications/callRecords/$entity",
  "version": 1,
  "type": "peerToPeer",
  "modalities": [
    "audio"
  ],
  "lastModifiedDateTime": "2020-02-25T19:00:24.582757Z",
  "startDateTime": "2020-02-25T18:52:21.2169889Z",
  "endDateTime": "2020-02-25T18:52:46.7640013Z",
  "id": "e523d2ed-2966-4b6b-925b-754a88034cc5",
  "organizer_v2@odata.context":
"https://graph.microsoft.com/v1.0/$metadata#communications/callRecords('e523d2ed-
2966-4b6b-925b-754a88034cc5')/organizer_v2/$entity",
  "organizer_v2": {
    "id": "821809f5-0000-0000-0000-3b5136c0e777",
    "identity": {
      "user": {
        "id": "821809f5-0000-0000-0000-3b5136c0e777",
        "displayName": "Abbie Wilkins",
        "tenantId": "dc368399-474c-4d40-900c-6265431fd81f"
      }
    }
  },
  "participants_v2@odata.context":
"https://graph.microsoft.com/v1.0/$metadata#communications/callRecords('e523d2ed-
2966-4b6b-925b-754a88034cc5')/participants_v2/$entity"
}
```

Response Schema

Table 1:The table describes each item in the response.

Element	Type	Description
id	String	A unique identifier for the call, which also serves as the identifier for the call record.
version	Int64	Represents the version of the call record, which increases each time the system generates a new call record. The higher the version of a call record, the more data it contains.
type	Enum	Indicates the type of the call. Values are unknown, `groupCall`, `peerToPeer`, `unknownFutureValue`.
modalities	Array[String]	List of all the modalities used in the call. Values are unknown, `audio`, `video`, `videoBasedScreenSharing`, `data`, `screenSharing`, `unknownFutureValue`.
lastModifiedDateTime	String (timestamp)	UTC time when the call record was created.
startDateTime	String (timestamp)	UTC time when the first user joined the call.
endDateTime	String (timestamp)	UTC time when the last user left the call.
organizer_v2	Object	Information about the call organizer.
organizer_v2/id	String	Unique identifier for the call organizer.
organizer_v2/ identity	Object	The identity of the call organizer.
organizer_v2/ identity/user	Object	The user associated with the organizer.
organizer_v2/ identity/user/id	String	The organizer's unique identifier (object id).
organizer_v2/ identity/user/ displayName	String	The organizer’s name.
organizer_v2/ identity/user/ tenantId	String	The organizer’s tenant id (organization id).
sessions	Array[Object]	List of sessions involved in the call. Peer-to-peer calls usually have one session, while group calls have at least one session per participant.
sessions[0]	Object	Represents a user-user communication of a per-to-peer call or a user-server communication in the case of a conference call.
participants_v2	Array	List of distinct participants in the call.
participants_v2[0]	Object	Represents the identity of a participant attended a call record.
joinWebUrl	String	Meeting URL associated to the call.
sessionsNextLink	String	The link to access the next page of `sessions`.
participantsNextLink	String	The link to access the next page of `participants_v2`.

Multiple Request Examples

Table 2:The table provides example uses and their corresponding URL with the query parameters.

URL	Usage
GET */communications/callRecords/{id}*	Retrieves call diagnostic information by call id.
GET */communications/callRecords/{id}?$select=type,startDateTime,endDateTime*	Retrieves call type, start and end time of the call by its id.
GET */communications/callRecords/{id}/sessions*	Retrieves a list of sessions associated with a specific call record.
GET */communications/callRecords/{id}/sessions?$expand=segments*	Retrieves a list of sessions associated with a specific call record, including the segments within each session.
GET */communications/callRecords/{id}?$expand=sessions*	Retrieves a call record by its id, including the sessions associated with the call record.
GET */communications/callRecords/{id}?$expand=sessions($expand=segments)*	Retrieves a call record by its id, including the sessions and segments within each session.
GET */communications/callRecords/{id}/participants_v2*	Retrieves a list of participants attended a peer to peer or a meeting call.
GET */communications/callRecords/{id}/?$expand=participants_v2*	Retrieves a call record by its id, including the participants associated with the call.
GET */communications/callRecords*	Retrieves a list of call records that occurred in the tenant organization for the past 30 days.
GET *communications/callRecords?$filter=participants_v2/any(p:p/id eq '{participantId}')*	Retrieves call records of the calls that occurred in the organization tenant over the past 30 days and include a specific participant id.
GET */communications/callRecords?$filter=startDateTime ge {startDate} and startDateTime lt {endDate}*	Retrieves call records that occurred in the organization tenant within a specified date range.
GET */communications/callRecords?$filter=startDateTime ge {startDate} and startDateTime lt {endDate} and participants_v2/any(p:p/id eq '{participantId}')*	Retrieves call records of the calls that occurred in the organization tenant within a specified date range for a specific participant id.

TUTORIAL: STEP-BY-STEP CALL RECORD SCENARIO INTERPRETATION

This tutorial will guide you through interpreting a call record received from the Graph Call Records API. You will use an example call record to understand each component and what it signifies.

Call Record Example

Note: *The given JSON example is not complete and is provided just for the sake of this tutorial.*


```
{
  "version": 1,
  "type": "peerToPeer",
  "modalities": [ "audio" ],
  "startDateTime": "2020-02-25T18:52:21.2169889Z",
  "endDateTime": "2020-02-25T18:52:46.7640013Z",
  "id": "e523d2ed-2966-4b6b-925b-754a88034cc5",
  "organizer_v2": {
    "identity": {
      "user": {
        "displayName": "Abbie Wilkins",
        "tenantId": "dc368399-474c-4d40-900c-6265431fd81f"
      }
    }
  },
  "sessions": [
    {
      "caller": {
          "platform": "android", "productFamily": "teams"
        },
        "associatedIdentity": {
          "displayName": "Abbie Wilkins",
          "tenantId": "dc368399-474c-4d40-900c-6265431fd81f",
        }
      },
      "callee": {
        "userAgent": {
          "platform": "windows", "productFamily": "teams"
        },
        "associatedIdentity": {
          "displayName": "Owen Franklin",
          "tenantId": "dc368399-474c-4d40-900c-6265431fd81f",
        },
        "feedback": { "rating": "poor" }
      },
      "segments": [
        {
          "media": [
            {
              "label": "main-audio",
              "callerDevice": {
                "receivedSignalLevel": -10
              },
              "streams": [
                {
                  "streamDirection": "callerToCallee",
                  "averageJitter": "PT0.016S"
                }
        ...
```

Call Record Scenario Interpretation

In this section, you will break down the call record into its individual components and find out what each part represents. This will help you understand the context, participants, and any potential issues with the call.

Property: type – [shows call type]
- **Value**: *peerToPeer*
- **Interpretation**: The call was between two people, one calling the other.

Property: modalities – [shows call modality]
- **Value**: *["audio"]*
- **Interpretation**: The call was audio-only, with no video or screen sharing.

Property: organizer_v2 – [shows call organizer]
- **Value**: *{"displayName": "Abbie Wilkins", "tenantId": "dc368399-474c-4d40-900c-6265431fd81f"}*
- **Interpretation**: Abbie Wilkins organized the call, and she belongs to the tenant organization with ID dc368399-474c-4d40-900c-6265431fd81f.

Properties: startDateTime, endDateTime – [start and end time; helps to calculate call duration]
- *Values*: "2020-02-25T18:52:21.2169889Z", "2020-02-25T18:52:46.7640013Z"
- *Interpretation*: The call started at 18:52:21 and ended at 18:52:46, lasting 25 seconds.

Property: sessions – [shows interaction between users]
- **Property: caller** – [initiator of the peer-to-peer call]
  - **Value***: {"displayName": "Abbie Wilkins", "platform": "android", "productFamily": "teams"}*
  - **Interpretation**: Abbie Wilkins joined the call from an Android device using the Teams client.
- **Property: callee** – [a person who answered the call]
  - **Value***: {"displayName": "Owen Franklin", "platform": "windows", "productFamily": "teams", "feedback": {"rating": "poor"}}*
  - **Interpretation**: Owen Franklin joined from a Windows device using the Teams client and rated the call as poor.

Property: segments/media – [shows information about media involved to the call]
- **Value:** *{"label": "main-audio", "callerDevice": {"receivedSignalLevel": -10}, "streams": [{"streamDirection": "callerToCallee", "averageJitter": "PT0.016S"}]}*
- **Interpretation:**
  - **Received Signal Level:** The signal strength was -10, indicating weak audio signals.
  - **Average Jitter:** The jitter was 0.016 seconds, which could have caused audio instability.

Conclusion

By analyzing each component of the call record, you can understand the call's context, participants, and potential issues. In this example, the poor rating by Owen Franklin was due to weak audio signals and jitter.

PRODUCT MANUAL: HOW TO RETRIEVE ALL PARTICIPANTS WHO ATTENDED A CALL

There are multiple options to collect all participants who attended a call.

Option 1: participants_v2 API

Use the List List participants_v2 API to collect the full list of participants for a requested call id.

Option 2: participants property

Use the Get callRecord API. Retrieve the *participants* property from a call record response to collect the participants who attended the call.

Please **note** that the *participants* property can contain a maximum of 60 participants because they are aggregated from the first 60 sessions. 60 is the maximum allowed page size for the returned sessions. If your call had more than 60 participants, the list will be incomplete.

Option 3: custom participants list

To see all participants, implement a custom solution to build the participants list from the call records data:

1. Call Get callRecord API with expanded sessions endpoint:

   /communications/callRecords/{id}?$expand=sessions

2. Read the *sessions@odata.nextLink* property value from the response. For example,

   https://graph.microsoft.com/v1.0/communications/callRecords/{callId}/sessions?

   $skiptoken={abc}

3. Repeat the next steps until the *sessions@odata.nextLink* is empty or null
   a. Send the GET request to the *sessions@odata.nextLink* URL
   b. Read *caller* property from each *session* on the received *sessions* list
   c. Build the participants list based on the caller identity by collecting id, displayName, and other properties you require to collect for a participant

HIGH-LEVEL ARCHITECTURE AND OUTAGE RESPONSE GUIDE

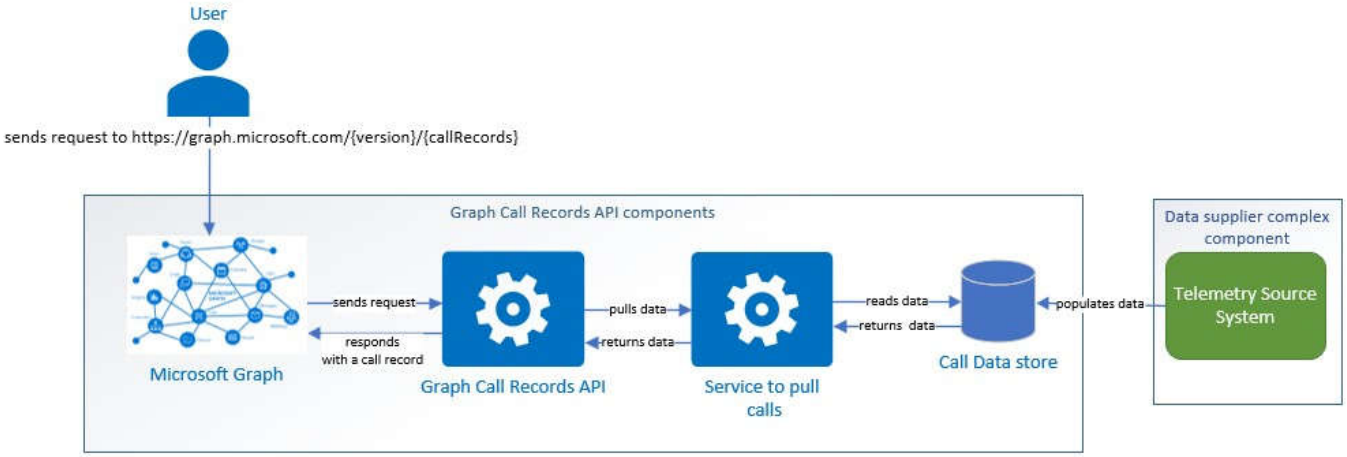
Purpose Statement

This document explains to administrators the high-level architecture design of the Microsoft Graph Call Records API. The document serves as an informational overview to help administrators understand how Graph Call Records API service works internally, enabling them to better comprehend the root cause of bottlenecks and respond to the reported outages.

Architecture Overview

The Graph Call Records API is designed to provide diagnostic call information to Microsoft Teams administrators. Image *Figure 1* describes the high-level flow to fetch a call record.

Figure 1: Graph Call Records API High-Level Architecture



The architecture involves several key components and processes.

Telemetry Source System

The Telemetry Source System is an Azure-based infrastructure that collects, processes, and organizes telemetry data from clients and services. It handles billions of daily events, enriching them with relevant metadata. This system builds call data from received telemetry and stores it in a high-throughput data store for quick retrieval. As a data supplier for the Microsoft Graph Call Records API, it enables administrators and developers to gain insights into communication patterns, troubleshoot issues, and optimize call performance within their organization.

Graph Call Records API flow

The process of serving Graph Call Records API:

1. **User Request:**
   Users make requests to the Graph Call Records API through tools like Postman or automated applications. The request is sent to
   {Method} https://graph.microsoft.com/{version}/communications/callRecords?{query-parameters}
2. **Microsoft Graph**:
   Microsoft Graph is a comprehensive platform that hosts various APIs. When a user makes a request to the Call Records API, Microsoft Graph redirects it to the dedicated Call Records API service.
3. **Graph Call Records API Service**:
   This service processes the request and calls an inner service to pull the requested call data.
4. **Service to pull data**:
   The inner service pulls the requested call data from the Azure-based Call Data Store. The Call Data Store contains data about the call, populated by the Telemetry Source System.
5. **Response Construction**:
   The inner service sends the call data back to the Graph Call Records API service. The Graph Call Records API service formats the data into a call record and sends it back to Microsoft Graph. Microsoft Graph then responds to the user with the final call record.

Graph Call Records API Outages: Causes and Solutions

During outages reported by the Graph Call Records API, customers frequently encounter numerous 500 Service Unavailable or 404 Not Found errors. The errors significantly affect the overall availability of the service. Common ***causes*** of these outages include:

1. **Telemetry Source System Outages**: The intricacy of the Telemetry Source System may lead to delays or errors, affecting the performance of the Graph Call Records API and resulting in delays in data delivery to the Call Data Store, which subsequently impacts data delivery to the customers.
2. **Azure Outages**: The Telemetry Source System components and the Call Records API pipeline components rely on Azure. Any issues with Azure can lead to significant delays or errors, affecting the timely delivery and accuracy of data.
3. **Deployment Issues**: Faulty deployments of any components involved to the Graph Call Records API flow can affect the service's availability.

During the outages, customers experience significant data delays, though data loss is rare.

The recommendation to customers is to implement a ***retry mechanism*** for calling the Call Records API until the data is successfully delivered. However, excessive retries from multiple organization tenants during an outage can lead to even longer delays due to system overload. Therefore, while a retry mechanism is recommended, administrators should also refer to the provided workaround steps on the outage ticket, available in the "Health" tab of the Teams Admin Portal for further assistance.

FAQ: FREQUENTLY ASKED QUESTIONS

What is the Microsoft Graph Call Records API?

Microsoft Graph Call Records API (also known as Call Records API) offers usage and diagnostic insights for calls made within a business organization, called a tenant, using Microsoft Teams or Skype for Business. Tenants can leverage the call records API to subscribe to, list, and retrieve call records by their ids, as well as to look up calls for a participant. For more details, please refer to the Working with the call records API in Microsoft Graph.

How do I request a call record using the Graph Call Records API?

To request a call record, you need to make a GET request to the */communications/callRecords/{id}* endpoint, where *{id}* is the unique identifier of the call. Ensure you have the necessary *CallRecords.Read.All* permission and include the Authorization header with a valid bearer token. For more details, please refer to the Get callRecord documentation.

What permissions are required to use the Call Records API?

To access call records in Microsoft Graph, the administrator must grant the *CallRecords.Read.All* permission. For more details, please refer to the Microsoft Graph permissions reference.

How long are call records retained?

Call records are retained for 30 days after a call ends.

Can I retrieve call records older than 30 days?

No, the Call Records API does not return call records older than 30 days. Requests for such records will result in a 404 Not Found response.

Where can I find the call ids of calls that occurred in my organization?

Four options to find and collect call ids exist.

Option 1: You can subscribe to change notifications feed. This will allow you to receive notifications containing call ids whenever a new call record is created.

Option 2: You can request List callRecords API that will supply you with the call ids list.

Option 3: If you are a Call Analytics customer, you can manually search for a call id in a user's history. However, there is no automated system available to retrieve all call ids from Call Analytics.

Option 4: You can use the *callChainId* of a Get call API to look up the call id once the call is completed. However, note that Get call API is not served by Graph Call Records API and in some cases, the *callChainId* may differ from the call id of your call record. Therefore, this method is less preferred compared to other options.

Why does a call record have missing fields?

A call record can have missing fields due to delayed telemetry from a client. When new telemetry data becomes available, the system generates a new call record with the updated information and increments the *version* property number. If you are missing properties on your call record, please wait for the next version to become available.

If the new call record version does not arrive, please, open a Support Ticket the Graph Call Records API team.

When will my call record be available?

On average, Graph Call Records service generates a new call record and sends a notification within 15 minutes after a call ends. However, it can take up to 60 minutes for the service to make the call record available.

Why is my call record notification delayed?

It can take up to 60 minutes for the service to make the call record available. If you experience a longer delay, check for any reported outages by the Graph Call Records API team by navigating to the "Health" tab in the Teams Admin Portal. Additionally, you can open Support Ticket the Graph Call Records API team.

I had hundreds of users on a meeting call. Why do I see only 60 users in *participants* property of a call record?

When you request a call record, the *participants* property is populated based on participants from the first 60 session where 60 is the maximum page size allowed on sessions. For more details, please refer to the How to retrieve all participants who attended a call.

How to interpret/understand a scenario?

Please, refer to the step-by-step call record scenario interpretation tutorial.

Why do I receive a 404 Not Found error?

There are several reasons you might encounter a 404 Not Found error:

Recent Call: If the call was made within the last 60 minutes, the call record might not have been generated yet. Please wait and try again in an hour.

Old Call: If the call is older than 30 days, the Graph Call Records API returns a 404 Not Found error by design.

Other Issues: If neither of the recent or old call reasons apply, check for any reported outages by the Graph Call Records API team by navigating to the "Health" tab in the Teams Admin Portal

Additionally, you can open Support Ticket the Graph Call Records API team for the assistance.