Content Summary Data

This document describes how to use Conviva Connect Content Summary reports. The Content Summary report is a daily offline historical log (view Conv provides session-level information for every video play or attempted play in a given day.

In This Topic

Audience

The Content Summary report is beneficial to many critical business departments, but it's mainly used by:

- Business analysts
- Operations teams
- Research teams

> ***Note:*** *To include the Conviva Household ID in your content summary data or to receive the data in parquet format, reach out to your Conviva rep*
>
> *Content summary now supports traffic rule accounts. You can now select traffic rule accounts (along with the preexisting non traffic rule accou the content summary data pipelines. For more details, contact Conviva Support.*

How to use Conviva Connect Fields?

We have summarized all the fields in a table, please review the Conviva Session Summary Schema at the end of this document.

You can use the Conviva Connect field data to:

- Filter against a particular metadata field in the Conviva Connect file to identify issues or patterns in a set of sessions.
- Filter against a particular Conviva Connect metric across one or many days.
- Calculate key experience and engagement metrics (refer to the Metrics section).
- Define and shape business KPIs (refer to the KPIs section).
- Map a user journey: You can use a common session ID across multiple data sources (from Conviva Connect and other logs or feeds) to track the use systems such as site or content discovery, page load, player load, CMS request, DRM request, ad request, CDN request, playback experience, next pag app crashes.
- Analyze viewer usage or consumption patterns: By tracking how users are engaging with content, you can research trends and patterns within viewin can be used in content licensing or production decisions.

You can also define custom metadata to focus your queries on your specific business needs. This processed information is intended for analysis using

> ***Note:*** *Most sensor integrations include ads delivered with SSAI (stitched ads) in the video content and metrics, and the content summary data. Events and Metadata section in the sensor integration instructions for more details about ad handling.*

Get the Conviva Connect Reports

The supported delivery destinations are:

- Google Cloud Storage Bucket
- Amazon S3 Bucket
- SFTP

The file path is:

[c3 account name]/[file type]/[date of data]/part-*

Data is delivered in multi-part files; part-* is used to enumerate the files.

The Conviva Connect file name is in the format:

```
DailyContentSessionLog_<CUSTOMER_NAME>_<YYYY-MM-DD>.csv
```

> ***Note:*** ByHousehold *is appended to the file name if the household ID column is included.*

Tools to Use

Typically, users load Conviva Connect into a database such as MySQL or into analytics tools such as Tableau or Micro Strategy to analyze the data and c sets. Refer to the sample SQL queries section for examples.

> ***Note:*** *When parsing legacy SSD and Connect files with .CSV data, any double quotes and commas passed to Conviva in the data columns are p prevent parsing errors and require percent-decoding. A sample Python decoding script in Spark is shown below. Conviva SSD and Connect files process double quotes and commas passed to Conviva without this decoding.*

In This Topic

```
def decode_pct_encoding(s):
    import urllib
    if s is None:
        return "Unknown"
    else:
         return urllib.parse.unquote_plus(s)
spark.udf.register("decode_pct_encoding", decode_pct_encoding)
```

Get Started with Content Summary Reports

1. Define Conviva Sessions

Each row in the Conviva Connect file describes a video session. Conviva defines a Session as an instance of an attempt to play a video; the attempt can action or an implicit automatic player action. In Video on Demand (VOD), the attempt would typically align with a specific asset; in linear live, the attempt multiple played assets within the same session.

Each session in Conviva Connect has a `ConvivaSessionID`, which is a string consisting of five segments. For example,

*8025681:1708681910:125006299:1630859473:*4109428019

The first four segments (in italic) represent the Client ID, which is assigned by Conviva to each device. The fifth segment (underlined) represents the Sess

A session that is suspended and resumed will appear as different rows with the same `ConvivaSessionID`. One row will have the session start time; th the session resume time(s). In this case, the primary key to process the sessions is *Conviva Session ID + Session Start Time*.

2. Define Lifetime Sessions

A Conviva Connect file contains snapshots of each lifetime session. A lifetime session starts on or before the pipeline end date and terminates on the pi

Example

Consider three lifetime sessions (with abbreviated IDs) from a Conviva Connect file with a December 16 pipeline end date:

a. First session (ID: 1001) starts and ends on December 16.

b. Second session (1002) starts on December 16 and crosses the midnight boundary into the next day (indicated by `EndedStatus = 0`).

c. Third row (ID: 1003) starts on December 15 and ends on December 16.

Three lifetime metrics are shown: *Playing Time*, *Connection Induced Rebuffering Ratio (CIRR)*, and *Attempt*. Note that because these are lifetime sessi 1003) includes *Playing Time* prior to December 16. All sessions are flagged as attempts because this status persists throughout the lifetime of the ses

Conviva Session ID	Start Time	End Time	Playing Time (seconds)	CIRR	Attempt	Ended Status
1001	2022-12-16 18:38:10	2022-12-16 18:59:10	1250	10	true	1
1002	2022-12-16 23:59:37	2022-12-16 23:59:59	20	0	true	0
1003	2022-12-15 23:54:00	2022-12-16 00:10:00	950	5	true	1

3. Lifetime versus Interval Sessions

Conviva Connect's lifetime sessions are different from Conviva Video that use interval sessions, which can be one minute, one hour, or longer. Because o and metric computations from Conviva Connect does not match the data and metric computations with ConvivaVideo. Refer to the Frequently Asked Qu details. To calculate interval metrics using Conviva Connect lifetime session data, refer to examples given in the Calculating Interval (day) Metrics sectio

Metrics List

Core Metrics	Attempts Plays

Engagement Metrics	Ended Plays Unique Devices Minutes Ended Play Average % Complete
Quality of Experience (QoE) Metrics	Video Startup Failures (VSF) Video Startup Failures Business (VSF-B) Video Startup Failures Technical (VSF-T) Video Playback Failures (VPF) Video Playback Failures Business (VPF-B) Video Playback Failures Technical (VPF-T) Exits Before Video Start (EBVS) Rebuffering Ratio Connection Induced Rebuffering Ratio Average Peak Bitrate Avg Average Bitrate Session Duration Bandwidth Video Startup Time Video Restart Time Micro Playing Time Micro Playing Interrupts Micro Buffering Time Micro Buffering Interrupts Long Rebuffering Time Long Rebuffering Interrupts Abandonment Paused Time Paused Ratio Last Playhead Time Bitrate Switches CIR Related Exits

In This Topic

Metric Definitions and Calculations

In most cases, lifetime metrics at the session level are provided in the Conviva Connect dataset. In cases where lifetime metrics at the session level are includes examples of the formulas and SQL code. The section below also shows how to aggregate the lifetime metrics across sessions. The SQL code i Connect date with December 16 (2022) pipeline end date.

Core Metrics

1. Attempts

Definition: An attempt is initiated when a viewer clicks play or a video auto-plays. An attempt can result in a successful play, or an early termination due t *(VSF)* or *Exits Before Video Start (EBVS)*. In Conviva Connect, each unique `ConvivaSessionID` + `StartTimeUnix` is an attempt. Across all sessions, t `Attempts` is:

$$\boxed{\text{Attempts} = \text{Count}(\text{Unique}(\text{ConvivaSessionID}, \text{StartTimeUnix}))}$$

```
SELECT
    COUNT(*) as Attempts
FROM
    Demo
```

2. Plays

Definition: A *Play* is a session with *Playing Time* greater than zero, which indicates that at least one video frame was played during the session lifetime.

There are two ways to aggregate the *Plays*. The first is a count of all sessions with `PlayingTime` > 0 milliseconds:

$$\boxed{\text{Plays} = \sum(\text{PlayingTime} > 0)}$$

```
SELECT
    COUNT(*) as Plays
FROM
    Demo
WHERE
    PlayingTime > 0
```

The second way is to count all session attempts that didn’t have a failure like VSF or EBVS:

$$\boxed{\text{Plays} = \text{Count}(\text{Attempt}) - \text{Count}(\text{VSF}) - \text{Count}(\text{EBVS})}$$

In some cases, the player may report a joined state, but the user might have exited before the first video frame was played. Joined time is indicated by S session joined but we don't have the actual join time, then we display -3 for `StartupTime`. Therefore, to aggregate Plays, you can include sessions wher and `PlayingTime >= 0`. See also What does VST = -3 stand for? in the FAQ section.

$$\boxed{\text{Plays} = \text{Count}(\text{StartupTime} \geq 0 | (\text{StartupTime} = -3\ \&\ \text{PlayingTime} \geq 0))}$$

In This Topic

```
-- Alternative method
SELECT
    COUNT(*) as Plays
FROM
    Demo
WHERE
    StartupTime > 0
or  (
        StartupTime = -3
    and PlayingTime >= 0)
```

QoE Metrics

3. Video Startup Failures (VSF)

Definition: VSF indicates if a video failed to play due to a startup error. This metric is available in Conviva Connect but is called `StartupError`. You can errors in the `ErrorList` field.

Across all sessions, the VSF % can be aggregated using the `Attempts` metric:

$$\boxed{\text{VideoStartFailure}\ \% = \frac{\sum(\text{StartupError} = \text{true})}{\text{Count}(\text{Attempt})} * 100}$$

```
SELECT
    (VSF / Attempts) * 100 as VSF_Perc
FROM
    (
        SELECT
            SUM (
                case
                WHEN
                    StartupError = 'true'
                then
                    1
                else
                    0
                END) as VSF,
            COUNT(*) as Attempts
        FROM
            Demo)
```

> ***Note:*** *Conviva lists all the VSF errors the player reports within 90 seconds of the failure in the order in which they were reported. Modifying the p reporting and error message text can help to clarify which error caused the failure.*

4. Video Startup Failures Business (VSF-B)

Definition: VSF-B indicates if a video failed to play due to a business error. Find the list of associated errors in the `VSFBusinessErrorList` field.

Across all sessions, the VSF-B % can be aggregated using the `Attempts` metric:

$$\boxed{\text{VideoStartFailure-Business}\ \% = \frac{\sum(\text{VideoStartFailure-Business} = \text{true})}{\text{Count}(\text{Attempt})} * 100}$$

```
SELECT
    (VSFB / Attempts) * 100 as VSFB_Perc
FROM
    (
        SELECT
            SUM (
                case
                WHEN
                    VSFBusiness = 'true'
                then
                    1
                else
                    0
                END) as VSFB,
            COUNT(*) as Attempts
        FROM
            Demo)
```

In This Topic

Note: *Conviva lists all the VSF-B errors the player reports within 90 seconds of the failure in the order in which they were reported. Modifying the reporting and error message text can help to clarify which error caused the failure.*

5. Video Startup Failures Technical (VSF-T)

Definition: VSF-T indicates if a video failed to play due to a technical error. Find the list of associated errors in the `VSFTechnicalErrorList` field.

Across all sessions, the VSF-T % can be aggregated using the `Attempts` metric:

$$\boxed{\text{VideoStartFailure-Technical } \% = \frac{\sum(\text{VideoStartFailure-Technical} = \text{true})}{\text{Count}(\text{Attempt})} * 100}$$

```
SELECT
    (VSFT / Attempts) * 100 as VSFT_Perc
FROM
    (
        SELECT
            SUM (
                case
                WHEN
                    VSFTechnical = 'true'
                then
                    1
                else
                    0
                END) as VSFT,
            COUNT(*) as Attempts
        FROM
            Demo)
```

Note: *Conviva lists all the VSF-T errors the player reports within 90 seconds of the failure in the order in which they were reported. Modifying the reporting and error message text can help to clarify which error caused the failure.*

6. Video Playback Failures (VPF)

Definition: VPF indicates if the video play terminates due to a playback error. VPFs are an important measurement of service quality and audience engag large percentage of plays terminate due to VPF. Find the list of associated errors in the `VPFErrorList` field.

Across all sessions, the VPF % can be aggregated using the `EndedPlays` metric:

$$\boxed{\text{VideoPlaybackFailure } \% = \frac{\sum(\text{VideoPlaybackFailure} = \ \text{true})}{\text{Count}(\text{EndedPlay})} * 100}$$

```
SELECT
    (VPF / EndedPlays) * 100 as VPF_Perc
FROM
    (
        SELECT
            SUM (
                case
                WHEN
                    VPF = 'true'
                then
                    1
                else
                    0
                END) as VPF,
            COUNT(*) as EndedPlays
        FROM
            Demo
        WHERE
            PlayingTime > 0
        and EndedStatus > 0)
```

Note: *Conviva lists all the VPF errors the player reports within 90 seconds of the failure in the order in which they were reported. Modifying the p reporting and error message text can help to clarify which error caused the failure.*

7. Video Playback Failures Business (VPF-B)

Definition: VPF-B indicates if the video play terminates due to a business playback error. VPF-Bs are an important measurement of service quality and au especially when a large percentage of plays terminate due to VPF-B. Find the list of associated errors in the `VPFBusinessErrorList` field.

Across all sessions, the VPF-B % can be aggregated using the `EndedPlays` metric:

$$\boxed{\text{VideoPlaybackFailure-Business \%} = \frac{\sum(\text{VideoPlaybackFailure-Business} = \text{true})}{\text{Count}(\text{EndedPlay})} * 100}$$

```
SELECT
    (VPFB / EndedPlays) * 100 as VPFB_Perc
FROM
    (
        SELECT
            SUM (
                case
                WHEN
                    VPFBusiness = 'true'
                then
                    1
                else
                    0
                END) as VPFB,
            COUNT(*) as EndedPlays
        FROM
            Demo
        WHERE
            PlayingTime > 0
        and EndedStatus > 0)
```

Note: *Conviva lists all the VPF-B errors the player reports within 90 seconds of the failure in the order in which they were reported. Modifying the reporting and error message text can help to clarify which error caused the failure.*

8. Video Playback Failures Technical (VPF-T)

Definition: VPF-T indicates if the video play terminated due to a technical playback error. VPF-Ts are an important measurement of service quality and au especially when a large percentage of plays terminate due to VPF-T. Find the list of associated errors in the `VPFTechnicalErrorList` field.

Across all sessions, the VPF-T % can be aggregated using the `EndedPlays` metric:

$$\boxed{\text{VideoPlaybackFailure-Technical \%} = \frac{\sum(\text{VideoPlaybackFailure-Technical} = \text{true})}{\text{Count}(\text{EndedPlay})} * 100}$$

```
SELECT
    (VPFT / EndedPlays) * 100 as VPFT_Perc
FROM
    (
        SELECT
            SUM (
                case
                WHEN
                    VPFTechnical = 'true'
                then
                    1
```

In This Topic

```
            else
                0
            END) as VPFT,
        COUNT(*) as EndedPlays
    FROM
        Demo
    WHERE
        PlayingTime > 0
    and EndedStatus > 0)
```

In This Topic

Note: *Conviva lists all the VPF-T errors the player reports within 90 seconds of the failure in the order in which they were reported. Modifying the reporting and error message text can help to clarify which error caused the failure.*

9. Exits Before Video Start (EBVS)

Definition: EBVS measures the number of viewing attempts that were terminated, typically by the viewer, before the video started. If an error is not genera attempt as an EBVS. An EBVS is similar to a VSF (failure to play video), but an EBVS does not include a descriptive error message.

The EBVS for each session needs to be computed from the Conviva Connect data.

$$\boxed{\mathbf{EBVS = True\ if\ (StartupTime = -1\ \&\ StartupError = 0\ \&\ PlayingTime = 0)}}$$

Across all sessions, the EBVS % can be aggregated using the `Attempts` metric:

$$\boxed{\mathbf{ExitsBeforeVideoStart\ \% = \frac{\sum(ExitsBeforeVideoStart = true)}{Count(Attempt)}}}$$

```
SELECT
    (EBVS / Attempts) * 100 as EBVS_Perc
FROM
    (
        SELECT
            SUM (
                CASE
                WHEN
                    StartupError    = 'false'
                    and PlayingTime = 0
                    and StartupTime = -1
                then
                    1
                else
                    0
                END) as EBVS,
            COUNT(*) as Attempts
        FROM
            Demo)
```

An alternative way to calculate EBVS is:

$$\boxed{\mathbf{EBVS = Count(Attempt)\ -\ Count(VSF)\ -\ Count(Play)}}$$

10. Rebuffering Ratio

Definition: *Rebuffering Ratio* shows the percentage of buffering during *Playing Time*. From this metric, we exclude the initial player startup buffering time frame).

Across all sessions, the *Rebuffering Ratio* % can be aggregated as:

$$\boxed{\mathbf{RebufferingRatio\ \% = \frac{\sum BufferingTime}{\sum(PlayingTime + BufferingTime)} * 100}}$$

```
SELECT
    (TotalBufTime / (TotalBufTime + TotalPlayTime)) * 100 as RebufferingRatio_Perc
FROM
    (
        SELECT
            SUM(BufferingTime) as TotalBufTime,
            SUM(PlayingTime)   as TotalPlayTime
        FROM
            Demo
        WHERE
            PlayingTime > 0)
```

11. **Connection Induced Rebuffering Ratio (CIRR)**

Definition: Rebuffering occurs when video play must wait for the buffer to fill because there's insufficient buffered video. CIRR is rebuffering unrelated to potentially caused by network conditions.

In This Topic

Across all sessions, the CIRR % is calculated using the `ConnectionInducedRebufferingTime`, `BufferingTime`, and `PlayingTime` fields in Conv

$$\boxed{\text{CIRR \%} = \frac{\sum \text{ConnectionInducedRebufferingTime}}{\sum(\text{PlayingTime} + \text{BufferingTime})} * 100}$$

```
SELECT
    COUNT(*) as Plays
FROM
    Demo
WHERE
    StartupTime > 0
or  (
        StartupTime = -3
    and PlayingTime >= 0)
```

> ***Note:*** *The Zero CIRR Ended Plays % metric can be derived by identifying the sessions that played without CIRR and calculating that percentage c the total sessions that played.*

12. **Average Peak Bitrate**

Definition: The average bitrate in kilobytes per second of delivered content across the lifetime session as derived from the bandwidth attribute in the play players may not report the bandwidth attribute.

Across all sessions, the *Average Peak Bitrate* can be aggregated as:

$$\boxed{\text{AveragePeakBitrate} = \frac{\sum(\text{AverageBitrate} * \text{PlayingTimeSec})}{\sum \text{PlayingTimeSec}}}$$

```
-- Because bitrate is per second, convert PlayingTime to seconds
SELECT
    (TotalBits / TotalPlayTimeSec) as AggAveragePeakBitRate
FROM
    (
        SELECT
            SUM(AverageBitRate * PlayingTime / 1000) as TotalBits,
            SUM(PlayingTime    / 1000) TotalPlayTimeSec
        FROM
            Demo
        WHERE
            PlayingTime > 0)
```

13. **Avg Average Bitrate**

Definition: The average bitrate in kilobytes per second of delivered content across the lifetime session, as derived from the average bandwidth attribut file. It represents the time-weighted average bitrates played by the player. Because some manifest files do not report average bandwith, the *Avg Avera* have more zero values than the *Average Peak Bitrate* field.

Across all sessions, the Avg Average Bitrate can be aggregated as:

$$\boxed{\text{AvgAverageBitrate} = \frac{\sum(\text{AvgAverageBitrate} * \text{PlayingTimeSec})}{\sum \text{PlayingTimeSec}}}$$

```
-- Because bitrate is per second, convert PlayingTime to seconds
SELECT
    (TotalBits / TotalPlayTimeSec) as AggAvgAverageBitRate
FROM
    (
        SELECT
            SUM(AvgAverageBitRate * PlayingTime / 1000) as TotalBits,
            SUM(PlayingTime / 1000) TotalPlayTimeSec
        FROM
            Demo
        WHERE
            PlayingTime > 0)
```

In This Topic

14. Session Duration

Definition: *Session Duration* is the total time in seconds from the session start to the session end. It is calculated as:

$$\boxed{\textbf{Session Duration} = \textbf{EndTimeUnix} - \textbf{StartTimeUnix}}$$

```
SELECT
    EndTimeUnix - StartTimeUnix as SessionDuration
FROM
    Demo
```

15. Bandwidth

Definition: *Bandwidth* is the amount of data transferred in megabytes (MB) per second across the lifetime session.

Across all sessions, the *Bandwidth* can be aggregated as:

$$\boxed{\textbf{Bandwidth} = \sum\left(\frac{\textbf{PlayingTime(ms)}}{\textbf{1000}} * \frac{\textbf{AverageBitrate(kbps)}}{\textbf{8000}}\right)}$$

```
-- Since bitrate is in seconds, convert PlayingTime to seconds
-- Since Average bitrate is kilobytes, convert to megabytes
SELECT
    SUM(Bandwidth) / COUNT(*) as Bandwidth
FROM
    (
        SELECT
            PlayingTime,
            (PlayingTime / 1000) * (AverageBitRate / 8000) as Bandwidth
        FROM
            Demo
        WHERE
            PlayingTime > 0)
```

16. Video Startup Time (VST)

Definition: VST is the number of seconds between the start of the Conviva monitoring (i.e. Session Start) and the first played video frame. We exclude an playing ads.

Across all sessions, VST can be aggregated using the `StartUpTime` value:

$$\boxed{\textbf{VideoStartupTimeSec} = \frac{\sum\textbf{(VideoStartupTime(ms)/1000)}}{\sum\textbf{(PlayingTime>0 \& StartupTime>=0)}}}$$

```
SELECT
    ROUND(TotalStartupTimeSec / Plays, 2) as StartupTimeSec
FROM
    (
        SELECT
            SUM(StartupTime) / 1000 as TotalStartupTimeSec,
            COUNT(*)                as Plays
        FROM
            Demo
        WHERE
            StartUpTime >= 0)
```

17. Video Restart Time (VRT)

Definition: VRT is the amount of time in seconds after seeking is completed by the user until the video begins playing.

Across all sessions, the VRT can be aggregated as:

$$\boxed{\mathrm{VideoRestartTimeSec} = \frac{\sum(\mathrm{VideoRestartTime(ms)}/1000)}{\sum(\mathrm{PlayingTime}>0\ \&\ \mathrm{RestartTime}\neq -1)}}$$

In This Topic

```
SELECT
    ROUND(TotalRestartTimeSec / Plays, 2) as RestartTimeSec
FROM
    (
        SELECT
            SUM(VideoRestartTime) / 1000 as TotalRestartTimeSec,
            COUNT(*)                     as Plays
        FROM
            Demo
        WHERE
            PlayingTime      > 0
        AND VideoRestartTime != -1 )
```

18. **Micro Playing Time**

    Definition: The total time in milliseconds that a session spent in continuous play time that is less than 200 milliseconds. Sometimes, the player report time is excluded from the *Playing Time*.

Across all sessions, the *Micro Playing Time* can be aggregated as:

$$\boxed{\mathrm{MicroplayingTimeMs} = \sum(\mathrm{MicroplayingTime})}$$

```
SELECT
    SUM(MicroplayingTime) as MicroplayingTimeMs
FROM
    Demo
WHERE
    PlayingTime > 0
```

19. **Micro Playing Interrupts**

    Definition: The total number of times a session spent in continuous play time that is less than 200 milliseconds. Sometimes, the player reports false p excluded from the *Playing Time*.

Across all sessions, *Micro Playing Interrupts* can be aggregated as:

$$\boxed{\mathrm{MicroplayingInterrupts} = \sum(\mathrm{MicroplayingInterrupts} = \mathrm{true})}$$

```
SELECT
    SUM(MicroplayingInterruptions) as MicroplayingInterrupts
FROM
    Demo
WHERE
    PlayingTime > 0
```

20. **Micro Buffering Time**

    Definition: The total time in milliseconds that a session spent in continuous buffering that is less than 200 milliseconds. Micro buffering could result i playback; this is not excluded from the session's buffering time.

Across all sessions, *Micro Buffering Time* can be aggregated as:

$$\boxed{\mathrm{MicroBufferingTimeMs} = \sum(\mathrm{MicroBufferingTime})}$$

```
SELECT
    SUM(MicroBufferingTime) as MicroBufferingTimeMs
FROM
```

```
    Demo
WHERE
    PlayingTime > 0
```

21. **Micro Buffering Interrupts**

Definition: The total count of number of times a session spent in continuous buffering that is less than 200 milliseconds. Micro buffering could result
playback; this is not excluded from the session's buffering time.

In This Topic

Across all sessions, *Micro Buffering Interrupts* can be aggregated as:

$$\boxed{\text{MicroBufferingInterrupts} = \sum(\text{MicroBufferingInterrupts} = \text{true})}$$

```
SELECT
    SUM(MicroBufferingInterruptions) as MicroBufferingInterrupts
FROM
    Demo
WHERE
    PlayingTime > 0
```

22. **Long Rebuffering Time**

Definition: The total time in milliseconds that a session spent in continuous buffering that is more than 90 seconds. Long buffering occurs when a pla
state; this is excluded from the session's buffering time.

Across all sessions, *Long Rebuffering Time* can be aggregated as:

$$\boxed{\text{LongRebufferingTimeMs} = \sum(\text{LongRebufferingTime})}$$

```
SELECT
    SUM(LongRebufferingTime) as LongRebufferingTimeMs
FROM
    Demo
WHERE
    PlayingTime > 0
```

23. **Long Rebuffering Interrupts**

Definition: The number of times a session spent in continuous buffering that is more than 90 seconds. Long buffering occurs when a player is stuck in
excluded from the session's buffering time.

Across all sessions, *Long Rebuffering Interrupts* can be aggregated as:

$$\boxed{\text{LongBufferingInterrupts} = \sum(\text{LongBufferingInterrupts} = \text{true})}$$

```
SELECT
    SUM(LongRebufferingInterruptions) as LongRebufferingInterrupts
FROM
    Demo
WHERE
    PlayingTime > 0
```

24. **Abandonment**

Definition: *Abandonment* occurs when a viewer exits a video before the video start with a wait time greater than 10 seconds (SPI setting defined as Good
setting defined as Best). At the session level, the *Abandonment* metric can be calculated as:

$$\boxed{\begin{aligned}\text{Abandonment} = \text{True if }(&\text{StartupTime} = -1 \ \& \ \text{StartupError} = 0 \ \& \\ &\text{PlayingTime} = 0 \ \& \ \text{SessionDurationSec} > 10)\end{aligned}}$$

```
SELECT
    CASE
    WHEN
```

```
        (StartUpTime           = -1
        and StartupError       = 0
        and PlayingTime        = 0
        and SessionDurationSec > 10)
    then
        1
    else
        0
    END as Abandonment10Sec
FROM
    (
        SELECT
            StartUpTime ,
            StartupError,
            PlayingTime ,
            (EndTimeUnix - StartTimeUnix) as SessionDurationSec
        FROM
            Demo)
```

In This Topic

Across all sessions, *Abandonment %* can be aggregated as:

$$\boxed{\text{Abandonment \%} = \frac{\sum(\text{Abandonment} = \text{true})}{\text{Count(Attempt)}} * 100}$$

```
SELECT
    (SUM(Abandonment10Sec) / Count(*)) * 100 as Abandonment10Sec_Perc
FROM
    (
        SELECT
            CASE
            WHEN
                (StartUpTime           = -1
                and StartupError       = 0
                and PlayingTime        = 0
                and SessionDurationSec > 10)
            then
                1
            else
                0
            END as Abandonment10Sec
        FROM
            (
                SELECT
                    StartUpTime ,
                    StartupError,
                    PlayingTime ,
                    (EndTimeUnix - StartTimeUnix) as SessionDurationSec
                FROM
                    Demo) )
```

A second approach more closely matches the SPI Streams calculation on Video. This calculation excludes from the denominator sessions that:

a) exited before the video started and the viewer did not wait a significant time (e.g. 8 seconds).

b) sessions impacted by business logic errors.

```
SELECT
    (SUM(Abandonment10Sec) / (COUNT(*) - SUM(BusinessError))) * 100 as Abandonment_Perc
FROM
    (
        SELECT
            CASE
            WHEN
                (StartUpTime           = -1
                and StartupError       = 0
                and PlayingTime        = 0
                and SessionDurationSec > 10)
            then
                1
            else
                0
            END as Abandonment10Sec,
            CASE
            WHEN
                (SessionDurationSec < 8
                or VSFBusiness      = true
                or VPFBusiness      = true)
            then
                1
            END as BusinessError
        FROM
            (
                SELECT
                    StartUpTime                                        ,
                    StartupError                                       ,
                    PlayingTime                                        ,
                    (EndTimeUnix - StartTimeUnix) as SessionDurationSec,
                    VSFBusiness                                        ,
                    VPFBusiness
                FROM
                    Demo))
```

In This Topic

25. Paused Time

Definition: *Paused Time* shows the total time in milliseconds the viewer paused during the session. A pause occurs when the viewer hits the pause bu

Across all sessions, the *Paused Time* can be aggregated as:

$$\text{PausedTimeMs} = \frac{\sum \text{PausedTime}}{\text{Count(EndedPlay)}}$$

```
SELECT
    AVG(PausedTime) as AggPausedTimeMs
FROM
    Demo
```

26. Paused Ratio

Definition: *Paused Ratio* gives the paused time as a ratio of the total playing time, including rebuffering and pauses. This metric helps you understand time during your video sessions.

Across all sessions, the *Paused Ratio* can be aggregated using the `EndedPlays` metric:

$$\text{PausedRatio \%} = \frac{\sum \text{PausedTime}}{\sum (\text{PausedTime} + \text{PlayingTime} + \text{BufferingTime})} * 100$$

```
SELECT
    SUM(PausedTime)/SUM(PausedTime + PlayingTime + BufferingTime) as AggPausedRatio
FROM
    Demo
WHERE
    EndedStatus > 0
and PlayingTime > 0
```

27. Last Playhead Time

Definition: The time in milliseconds of the last playhead position. This metric is available only at the session level.

28. Bitrate Switches

Definition: A bitrate switch occurs whenever a change in bitrate is detected. The *Bitrate Switches* metric displays the number of the bitrate switches o

Across all sessions, the average number of *Bitrate Switches* can be aggregated using the `EndedPlays` metric:

$$\text{BitrateSwitches} = \frac{\sum \text{BitrateSwitches}}{\text{Count(EndedPlay)}}$$

```
SELECT
    AVG(NumBitrateSwitches) as AggNumBitRateSwitches
FROM
    Demo
WHERE
    EndedStatus > 0
and PlayingTime > 0
```

In This Topic

29. CIR Related Exits

Definition: An exit that occurs during connection induced rebuffering (and not due to seek) within 5 seconds before the session end.

Across all sessions, `CIRRelatedExits %` can be aggregated using the `EndedPlays` metric:

$$\boxed{\text{CIRRelatedExits \%} = \frac{\sum(\text{CIRRelatedExit} = \text{ true})}{\text{Count(EndedPlay)}} * 100}$$

```
SELECT
    (CIRRelatedExits / EndedPlays) * 100 as CIRRelatedExits_Perc
FROM
    ( SUM(
        CASE
        WHEN
            CIRRelatedExit = 'true'
        then
            1
        END) as CIRRelatedExits, COUNT(*) as EndedPlays FROM Demo WHERE EndedStatus > 0
    and PlayingTime > 0)
```

Engagement Metrics

30. Ended Plays

Definition: An *Ended Play* is a session with a *Play* (`PlayingTimeMs > 0`) that ended (`EndedStatus > 0`) during the selected time frame, in this case t
December 16) of the Conviva sessions data.

Across all sessions, the number of *Ended Plays* can be aggregated as:

$$\boxed{\text{Ended Plays} = \text{Count(PlayingTime} > 0 \text{ \& EndedStatus} > 0)}$$

```
SELECT
    Count(*) as EndedPlays
FROM
    Demo
WHERE
    PlayingTime > 0
and EndedStatus > 0
```

31. Unique Devices

Definition: *Unique Devices* counts the total number of devices that had any *Ended Plays* during the during the selected timeframe, in this case the pipelin
December 16) of the Conviva sessions data. A unique device is not equivalent to a unique person. If a person uses multiple devices, each device is count
there are more than one applications offered to the same device, the *Client ID* will be identified for each application separately. This *Client ID* is represent
components of the *Conviva Session ID*, as explained here.

Across all sessions, *Unique Devices* can be aggregated using the `EndedPlays` metric:

$$\boxed{\text{UniqueDevices} = \text{Count(Unique(ClientID \& EndedPlay} = \text{true))}}$$

```
SELECT
    COUNT(DISTINCT(CONCAT(ID[0], ":", ID[1], ":", ID[2], ":", ID[3]))) as UniqueDevices
FROM
    (
      -- first split sessionID into components, then construct clientID from first four
```

```
        SELECT
            SPLIT(ConvivaSessionID, ":") as ID
        FROM
            Demo
        WHERE
            PlayingTime > 0
        and EndedStatus > 0)
```

In This Topic

32. Minutes Ended Play

Definition: The *Minutes Ended Plays* metric is calculated by dividing the total amount of *Playing Time* minutes by the number of *Ended Plays* during the s
case the pipeline end date (for example, December 16).

Across all sessions, *Minutes Ended Plays* can be aggregated using the `EndedPlays` metric:

$$\boxed{\text{PlayingTimeMins} = \frac{\sum(\text{PlayingTime(ms)}/60000)}{\text{Count(EndedPlay)}}}$$

```
SELECT
    TotalPlayTimeMin / TotalEndedPlays as MinuteEndedPlay
FROM
    (
        SELECT
            -- convert millisec to minutes
            SUM(PlayingTime / (1000 * 60)) as TotalPlayTimeMin,
            COUNT(*)                       as TotalEndedPlays
        FROM
            Demo
        WHERE
            PlayingTime > 0
        and EndedStatus > 0)
```

33. Average % Complete

Definition: *Average % Complete* shows the amount of viewed play duration compared with the total length of the content. A high % indicates a high level
with the asset, channel, and service.

$$\boxed{\text{Average \% Complete} = \frac{\sum \text{PlayingTime}}{\sum \text{ContentLength}} * 100}$$

The field `PercentageComplete` exists at the session level, which can be used to calculate the *Average % Complete* across all sessions.

```
SELECT
    AVG(PercentageComplete) as AvePercComplete
FROM
    Demo
WHERE
    PercentageComplete >= 0
```

Calculating Interval (day) Metrics

It is not always possible to directly calculate interval metrics from the Conviva Connect lifetime sessions. In this section, we show how to calculate inter
lifetime sessions using the example from the Get Started with Content Summary Reports section. In this dataset, the third session (ID: 1003) spans two
we cannot infer the exact *Playing Time* and *CIRR* for the December 16 day-interval (00:00:00 to 23:59:59) using the December 16 pipeline end date. One
dataset using the December 15 pipeline end date. For example, the following dataset has only one session (ID: 1003) because the other sessions start o

Conviva Session ID	Start Time	End Time	Playing Time (seconds)	CIRR	Attempt	Ended Status
1003	2022-12-15 23:54:00	2022-12-15 23:59:59	355	2	true	0

To calculate *Playing Time* and *CIRR* for the December 16 day-interval, subtract the respective lifetime metric ending on December 15 from the lifetime m
16. So for the December 16 day-interval, `intvPlayingTime = 950 - 355 = 595` and `intvCIRR = 5 - 2 = 3`. Similarly, for `ConvivaSessionI`

`intvAttempt = true` since both sessions started on this day whereas the third session (ID: 1003) did not, hence, `intvAttempt = false`.

Sample metrics for the December 16 day-interval:

Conviva Session ID	Start Time	End Time	intv Playing Time (seconds)	intv CIRR	intv Attempt	Ended Status
1001	2022-12-16 18:38:10	2022-12-16 18:59:10	1250	10	true	1
1002	2022-12-16 23:59:37	2022-12-16 23:59:59	20	0	true	0
1003	2022-12-15 23:54:00	2022-12-16 00:10:00	595	3	false	1

In This Topic

The SQL code shows one way to do this interval calculation, where `Demo16Dec` is the December 16 pipeline end date and `Demo15Dec` is the December

```
SELECT
    L.ConvivaSessionID                                               ,
    L.PlayingTime                                                    ,
    L.CIRR                                                           ,
    L.PlayingTime - IFNULL(P.PlayingTimePrior, 0) as IntvPlayingTime,
    L.CIRR        - IFNULL(P.CIRRPrior, 0)        as IntvCIRR
FROM
    Demo16Dec L
LEFT OUTER JOIN
    Demo15Dec P
ON
    (
        L.ConvivaSessionId = P.ConvivaSessionID)
```

To determine an attempt for the 16 December day-interval, we use the `Demo16Dec` data and the `StartTimeUnix` field:

```
SELECT
    CASE
        (
    WHEN
        FROM_UNIXTIME(StartTimeUnix, 'y-MM-dd') = "2022-12-16"
    then
        1
    else
        0
    END) as intvAttempt
FROM
    Demo
```

To calculate the interval-based metrics such as *Plays*, *Video Startup Failure*, *Video Restart Time*, and others, follow the same logic described above. Plea support if needed.

KPIs

Startup

- `Startuptime>= 0` AND `PlayingTime = (number)` AND no error code, corresponds to a successful play.
- `Startuptime= -1` AND `PlayingTime = 0` AND an error code, corresponds to a VSF.
- `Startuptime = -1` AND `PlayingTime = 0` AND no error code, corresponds to an EBVS.

Quality KPI: Successfully started sessions

- `StartupError = 0` and `BufferingTime > 0` => Pass
- `StartupError <> 0` or `PlayingTime < 1` => Fail

Quality KPI: Sessions with `StartupTime` below acceptable threshold

- `StartupTime <= Threshold` => Pass

- `StartupTime > Threshold` => Fail

Quality KPI: Sessions Played with an acceptable bitrate

- `AvgBitrate >= Threshold` => Pass
- `AvgBitrate < Threshold` => Fail

Quality KPI: Sessions Played with acceptable buffering ratio

- `(BufferingTime / BufferingTime + PlayingTime) ∗ 100 <= Threshold` => Pass
- `(BufferingTime / BufferingTime + PlayingTime) ∗ 100 > Threshold` => Fail

In This Topic

Conviva SPI Calculation

The Conviva SPI provides a visual indicator of your KPIs performance so you can quickly determine the number and percent of impacted streams and pe is impacted when it fails to meet one or more of the defined KPI settings.

Conviva formulates a unified streaming performance KPI based on the percentage of streaming sessions with good or best viewing experience. This KP Streaming Performance Index, and is based on the percentage of streams with:

- No errors (VSF-T or VPF-T)
- No or very low Rebuffering (using CIRR)
- Acceptable picture quality based on average bitrate for different screen sizes
- Acceptable Video Start Time
- No EBVS if the viewer was waiting a long time before exiting.

Conviva provides KPI settings based on Good and Best performance. You can also set custom KPI settings to match your specific performance goals.

Good Performance KPI Settings

Best Performance KPI Settings

Frequently Asked Questions

1. **Difference between Video dashboard data and Conviva Connect**

There are differences between the data that you see within Conviva Video vs what you see in the Conviva Connect SSD. This is by design, and there are a Conviva Connect and Conviva Video have differences:

a. Conviva Video uses real-time, 1-minute or 1-hour interval metrics whereas Conviva Connect SSD provides lifetime metrics and can provide a historical averages between the two will get closer the longer the window you use. That is, a 24-hour average results in a better match than a 1-hour average.

b. Although Conviva Video and Conviva Connect share the same session level sanitization rules, Conviva Video adds additional aggregation logic based The idea here is for the consumer of Session Source Data to apply aggregation logic based on the desired use case.

Pursuing an exact match between Conviva Video and offline Conviva Connect calculations is difficult and often inadvisable.

A Note on Heartbeats:

Conviva uses Heartbeats to monitor the video sessions.
Heartbeats continue to be sent while the video is buffering, unless the internet connection itself has been interrupted and the Conviva library and backen this case, the library will continue monitoring and will send a "catch-up" heartbeat when the connection is re-established.
If the video is paused or the connection is lost for more than 2 minutes, then no heartbeats will be sent and the Conviva back-end will automatically clear not count paused time, so if the viewer was paused for more than 2 minutes then the session will time out.
We still collect heartbeats during rebuffering, so this would not cause a session to timeout. However, if the player ends up in what we call a 'zombie state buffering) then that contributes to Conviva Video and Conviva Connect metrics. The only difference is that Conviva Video metrics cap buffering at 30 mir Connect has no such cap.

2. **What does VST = -3 stand for?**

VST=-3 means we don't have a specific Video Startup Time. Possible reasons include:

- The current session is the revived portion of an expired session. Sessions expire when there is 120 seconds of inactivity. The expired and revived sess session ID, so you can group the sessions by session id (Conviva SessionID). The session End flag of the expired session is set to 2 (Session End Statu portion has VST of -3.
- Conviva flags indicate that the *Startup Time* is not reliable.
- *Startup Time* exceeded 10 minutes and the VST is flagged as unknown.

3. **Why do I see unexpected ASCII characters in certain fields?**

Conviva Connect reports are delivered as CSV files. The Session Tags column in Conviva Connect is customer defined and can contain arbitrary strings v which could prevent parsing of CSV; therefore we use URI encoding to remove any offending characters. We recommend that you run URI decoding on th using various libraries that provide this decode functionality (available in all mainstream languages).

4. How do I stitch Conviva Connect across multiple days?

The data in Conviva Connect is already "stitched" for sessions that span across multiple days. Conviva Connect provides lifetime metrics, which means t spans across many days, the last session record contains the complete details of the metrics for that session. This is a quick mechanism to view Conviv across multiple sets. Therefore, when combining Conviva Connect sessions, overriding earlier sessions with the most recent session provides the most a mechanism to stitch session logs.

5. What is the difference between -1 and 0 in Percent Complete?

The value in Percentage Complete is rounded to the nearest integer value.
A value of 0 is a rounded value when percentage complete is less than 1%. This covers the occasions when the `StartupTime>0` but the video only play of time.
A value of -1 (usually for live content) indicates that we do not receive content length.

In This Topic

Conviva Content Session Summary Schema

> ***Note:*** *The Content Summary files contain new fields and metrics that were unavailable in the legacy SSD file. Also, several of the legacy SSD fiel renamed in the Content Summary schema for improved consistency.*

New Fields in Content Summary Schema	BrowserVersion ConnectionType DeviceHardwareType DeviceManufacture DeviceMarketingName DeviceModel DeviceName DeviceOSFamily DeviceOSVersion DeviceVendor DMA dt HouseholdIP	IPType IPV4 IPV6 LastPlayheadTime NumBitrateSwitches PausedRatio PauseTime PlayerFrameworkName PlayerFrameworkVersion PostalCode ReBufferingTime VSFTechnical
Renamed Fields	**SSD Field Name**	**Content Summary Field Name**
	Session End Status	EndedStatus
	Session End Time	EndTimeUnix
	Micro Rebuffering Interrupts	MicroBufferingInterruptions
	Micro Rebuffering Time	MicroBufferingTime
	Micro Playing Count	MicroPlayingInterruptions
	Percent (%) complete	PercentageComplete
	Start Time	StartTimeUnix

Content Session Summary Field Descriptions

> ***Note:*** *All INTEGER type fields can store 64-bit integer values.*

Field name	Type	Mode	
ViewerID	VARCHAR(128)	NULLABLE	Unique identifier of the viewer (sometimes called subscriber) watch personally identifiable information. The same ViewerId can have mu ***Note:*** *If Viewer ID is not available, in the legacy SSD report,*
AssetName	STRING	NULLABLE	The name of the viewed video asset

In This Topic

Field name	Type	Mode	
DeviceOS	STRING	NULLABLE	The operating system of the device
DMA	STRING	NULLABLE	The Designated Market Area or media region in which the session w
Postal Code	STRING	NULLABLE	A series of numbers used for postal delivery area identification. This
Country	STRING	NULLABLE	The country location where the content was watched.
State	VARCHAR(128)	NULLABLE	The state location where the content was watched
City	STRING	NULLABLE	The city location where the content was watched
ASN	STRING	NULLABLE	Autonomous System Number for the ISP from which the video was
ISP	STRING	NULLABLE	The name of the Internet Service Provider
StartTimeUnix	INTEGER	NULLABLE	The time when Conviva received the first heartbeat for the session.
StartTimeUnixMs	INTEGER	NULLABLE	The time when Conviva received the first heartbeat for the session.
StartupTime	INTEGER	NULLABLE	The time in milliseconds between the start of the Conviva monitorin -1 indicates an unsuccessful play (no startup time). -3 indicates the session connected, but the client didn't send us the
PlayingTime	INTEGER	NULLABLE	The amount of time in milliseconds when a player is actively display
ReBufferingTime	INTEGER	NULLABLE	The time between the video stalling during playback and the viewer
Interrupts	INTEGER	NULLABLE	The number of times the session was interrupted for rebuffering. If pausing and resuming a session without any buffering is not counte
AverageBitRate	INTEGER	NULLABLE	Average bitrate in kbps at which content was delivered during the se deliver bitrate information.
StartupError	SMALLINT	NULLABLE	If value = true, then the video played and there was no startup error. If value = false, then the video failed to play and there was a startup
SessionTags	RECORD[VARCHAR(64)/each key, VARCHAR(256)/each Value]	REPEATED	The custom player metadata that is defined during your Conviva int For CSV output file the value is in `String` format, where: • Key-value pairs are delimited by ampersand (&) • Key and value are separated by equals sign (=) For example, c3.cmp.0._id=da&c3.cmp.0._ver=1&c3.cluster.name=production&c3 For Parquet file format, the value is in `array<struct<_field1:s`
IPV4	VARCHAR(32)	NULLABLE	The public IP address of the viewer's video playing device in v4 vers
IPV6	VARCHAR(48)	NULLABLE	The public IP address of the viewer's video playing device in v6 vers
IPAddress	VARCHAR(48)	NULLABLE	The public IP address of the viewer's video playing device. The IP ac connections or the packet gateway for mobile connections. For European customers, due to legal/privacy reasons, the IP addres
IPType	STRING	NULLABLE	The type of the device's public IP address, such as, IPV4 Only, IPV6
CDN	VARCHAR(256)	NULLABLE	The CDN associated with the streaming session
Browser	STRING	NULLABLE	The browser used by the viewer's device."Non-Browser Apps" is sho
ConvivaSessionID	VARCHAR(128)	NULLABLE	The unique Conviva session identifier in format of five, colon-separa represents the Conviva Session ID. Example: Conviva session id value = "20048757:2397552430:41513

In This Topic

Field name	Type	Mode	
			Client ID = 20048757:2397552430:4151350518:1876058113 Session ID = 4487054
StreamURL	VARCHAR(2048)	NULLABLE	The URL of the video stream
ErrorList	[VARCHAR(1024)/each]	REPEATED	A list of fatal errors that occurred during this session, separated by Video Start (EBVS) occurred. For CSV output file, the ErrorList value is in `String` format. For Parquet file, the value is in `array<string>` format.
PercentageComplete	INTEGER	NULLABLE	The percentage of video content the viewer watched during a sessi length. % Complete is rounded to the nearest integer value. A value of -1 means we couldn't obtain content length (for example, A value of 0 means that the video did not start or that the Percentag
ConnectionInducedRebufferingTime	INTEGER	NULLABLE	The non-seek rebuffering time in milliseconds
VideoRestartTime	INTEGER	NULLABLE	The total time between the user's seek complete and the video repl
RejoinedCount	INTEGER	NULLABLE	Number of times the video rejoined after a user seek
VPF	SMALLINT	NULLABLE	Video Playback Failures (VPF) occurs when a fatal error causes a vi The field is set to TRUE if the session started successfully but ende
VPFErrorList	RECORD	REPEATED	Video Playback Failure Error list contains errors (including custom e For CSV output file, the VPFErrorList value is in `String` format. For Parquet file, the value is in `array<string>` format.
ContentLength	INTEGER	NULLABLE	The asset length in milliseconds. Applicable only for VOD traffic. Fo
EndedStatus	INTEGER	NULLABLE	An integer (0-5) showing the session status at the end of the day: 0 1 = Gracefully ended; the session ended with a session ended event 2 = Expired due to lack of heartbeat update; No heartbeat update wa 3 = Expired due to long buffering; the total session lifetime buffering 4 = Ended due to long pause; the session paused for a continuous p 5 = Ended due to continuous buffering; session was in a continuous
SessionEndedStatus	STRING		The state of the session when it was ended, such as, GracefulEnd, N
EndTimeUnix	INTEGER	NULLABLE	The time the last session heartbeat within the day was received. Th
EndTimeUnixMs	INTEGER	NULLABLE	The time the last session heartbeat within the day was received. Th
VSFBusiness	BOOLEAN	NULLABLE	Video Start Failures (VSF) Business measures whether the Attempt reported due to a business logic issue, such as usage limits.
VSFBusinessErrorList	RECORD	REPEATED	Video Start Failures (VSF) Business Error list contains errors (incluc For CSV output file, the VSFBusinessErrorList value is in `String` fo For Parquet file, the value is in `array<string>` format.
VSFTechnical	BOOLEAN	NULLABLE	Video Start Failures (VSF) Technical measures whether the Attempt reported due to a technical logic issue, such as prolonged buffering
VSFTechnicalErrorList	RECORD	REPEATED	Video Start Failures (VSF) Technical Error list contains errors (incluc For CSV output file, the VSFTechnicalErrorList value is in `String` fo For Parquet file, the value is in `array<string>` format.

Field name	Type	Mode	
VPFBusiness	SMALLINT	NULLABLE	Video Playback Failures (VPF) Business measures how often Attem such as usage limits
VPFBusinessErrorList	RECORD	REPEATED	Video Playback Failures (VPF) Business Error list contains errors (ir For CSV output file, the VPFBusinessErrorList value is in `String` fo For Parquet file, the value is in `array<string>` format.
VPFTechnical	BOOLEAN	NULLABLE	Video Playback Failures (VPF) Technical measures whether the Atte issue, such as prolonged buffering.
VPFTechnicalErrorList	RECORD	REPEATED	Video Playback Failures (VPF) Technical Error list contains errors (ir For CSV output file, the VPFTechnicalErrorList value is in `String` fo For Parquet file, the value is in `array<string>` format.
PauseTime	INTEGER	NULLABLE	Total pause time in milliseconds for a session.
CIRRInterruptCount	INTEGER		The number of plays with interrupts caused by connection induced
MicroPlayingTime	INTEGER	NULLABLE	The total time in milliseconds that a session spent in continuous pla
MicroPlayingInterruptions	INTEGER	NULLABLE	The total number of times a session spent in continuous play time t is excluded from the Playing Time.
MicroBufferingTime	INTEGER	NULLABLE	The total time in milliseconds that a session spent in continuous bu playback and this is not excluded from the session's buffering.
MicroBufferingInterruptions	INTEGER	NULLABLE	The total number of times a session spent in continuous buffering t buffering occurs and is not excluded from the session's buffering.
LongRebufferingTime	INTEGER	NULLABLE	The total time in milliseconds that a session spent in continuous bu buffering state. Long rebuffering is excluded from rebuffering time.
LongRebufferingInterruptions	INTEGER	NULLABLE	The total number of times a session spent in continuous buffering t buffering state. Long rebuffering this is excluded from rebuffering ti
LastCDNEdgeServer	STRING	NULLABLE	The IP address of the CDN Edge Server.
LastCDNGroupID	STRING	NULLABLE	The region or pop identifier of the CDN Edge Server.
ExitDuringPreRoll	SMALLINT	NULLABLE	A started session exited after a pre-roll ad break start was reported
AdRelatedRebuffering	INTEGER	NULLABLE	Rebuffering duration in milliseconds which started up to 60 seconds
RebufferingDuringAds	INTEGER	NULLABLE	Rebuffering duration in milliseconds happening during the ad playba
PausedRatio	FLOAT	NULLABLE	The paused time as a ratio of the total playing time, including rebuff
LastPlayheadTime	INTEGER	NULLABLE	The last play time, after which a pause, end, or expire event occurre
NumBitrateSwitches	INTEGER	NULLABLE	The number of the bitrate switches that occurred during the lifetime
AvgAverageBitRate	INTEGER	NULLABLE	The average bitrate (in kilobytes per second) across the lifetime ses time-weighted average bitrates played by the player.
CIRRelatedExit	SMALLINT	NULLABLE	A user initiated exit that occurred either during connection induced the session end.
HouseholdID	VARCHAR(128)	NULLABLE	Unique identifier for a Household
HouseholdIP	VARCHAR(48)	NULLABLE	The physical residence's IP address, where the devices are located. ***Note:*** *If a Household IP address is accessible in IPv4 and II ID, one with an IPv4 address and the other with an IPv6 adc To differentiate the records, Conviva advises to include bot pipeline*
DeviceHardwareType	STRING	NULLABLE	The type of your device hardware such as set top box, mobile phone

In This Topic

Field name	Type	Mode	
DeviceManufacture	STRING	NULLABLE	The manufacturer of the device from which the content was watche
DeviceMarketingName	STRING	NULLABLE	Marketing name of the device from which the content was watched
DeviceName	STRING	NULLABLE	Name of the device from which the content was watched, such as, /
DeviceOSVersion	STRING	NULLABLE	The version of the operating system used by the device
DeviceOSFamily	STRING	NULLABLE	The name of the operating system group, such as PlayStation for Pl
BrowserVersion	STRING	NULLABLE	The browser version of the device on which the content was watche
PlayerFrameworkName	STRING	NULLABLE	The name of the player framework used for video playback, for exar
PlayerFrameworkVersion	STRING	NULLABLE	The version of the framework used for video playback.
DeviceModel	STRING	NULLABLE	Model of the device, such as, iPad Pro 11-inch (2nd generation), EM
DeviceVendor	STRING	NULLABLE	Vendor of the device
ConnectionType	STRING	NULLABLE	The type of network connection used to consume content, for exam
DecisionBitrate	INTEGER	NULLABLE	The btirate associated with resource returned to Precision. A value
DecisionResource	STRING	NULLABLE	The internal resource returned to Precision.
DecisionResourceId	STRING	NULLABLE	The internal Conviva id for the component returning the result to Co
DecisionResourceResolved	STRING	NULLABLE	The resource being returned by Conviva Precision, for example Akar
pCoreCDN	STRING	NULLABLE	The CDN being returned by Conviva Precision, for example AKAMAI
dt	TIMESTAMP	NULLABLE	Starting hour of the hourly interval range, in the format <YYYY-MM-DI For example, 2024-05-21T15:00:00.000Z.

Note: *HouseholdID is available only for customers who have StreamID feature enabled.*

In This Topic