Adobe® Marketing Cloud

ClickStream Data Feeds

Contents

Clickstream Data Feeds Help

Important: *See What's New to learn about an upcoming column expansion that impacts all data feeds customers.*

Raw clickstream data that is collected from web sites, mobile apps, or is uploaded using web service APIs or data sources, is processed and stored in Adobe's data warehouse. This raw clickstream data forms the data set that is used by Adobe Analytics.

As a service, Adobe can deliver this raw data to the customer in a batched manner on a recurring daily or hourly delivery schedule. This service is called the "raw clickstream data feed", or just "data feed." Each data feed comprises the clickstream data for a single report suite.

To get started with clickstream data feeds, see *Configuring Data Feeds*.

For column reference for an existing feed, see *Clickstream Data Column Reference*.

Configuring Data Feeds

Data feeds are enabled by Customer Care and delivered using FTP.

This section provides an overview of data feed options and the one-time configuration process.

FTP File Delivery

Data feed data can be delivered to an Adobe or customer hosted FTP location.

If you select to have data uploaded to your FTP server, you must provide Adobe with the appropriate username, password, and upload path. You must implement your own process to manage disk space on the server, as Adobe does not delete any data from the server.

Delivery Formats and Contents

Daily: Data for each day is delivered after it is processed in a single zipped file, or in multiple zipped files each containing approximately 2 GB of uncompressed data. You receive a single delivery for each day.

Hourly: Data for each hour is delivered in a single zipped file that contains all data received during that hour. You receive 24 separate deliveries for each day, with each file delivered after the data for that hour is processed.

Note: *Due to the potential size of data feed zip files, make sure your ETL process uses a 64-bit zip utility.*

Hourly Data Feeds

It is important to understand that the term “hourly” describes the time frame of the data that is sent with each individual data export, and not the time frame in which the delivery occurs. Hourly data feeds are processed and delivered in a best-effort fashion. However, there are several factors that can impact the delivery time of an hourly data feed including:

- Report suite latency (i.e. unannounced spike in traffic)
- Upstream processing
- Peek and non-peak hours
- Internet connection speeds

For hourly data feeds the expectation is that 95% of the time the feed will deliver within six hours of the close of that hour’s worth of data.

Receiving an hourly data feed is different then receiving daily feed with multiple file delivery. When receiving hourly data feeds the data for each day is split into 24 files based on the data collected during that hour, and each file is delivered as soon as it is available. A daily feed that is delivered in multiple files is delivered once per day after the previous day's data is processed, and is spilt into 2GB increments based on the amount of data collected.

Data Backfills for Hourly Data Feeds

If you request data for earlier dates when setting up a new hourly data feed, data for dates more than 60 days ago might be delivered in daily format instead of hourly.

In this case, you will not receive 24 separate deliveries for these days, instead, you will receive a single delivery with a midnight timestamp that contains all of the data for that day. If you are requesting this type of backfill, Make sure your ETL is configured to process daily deliveries.

Multiple File Delivery

You can select single file or multiple file delivery when the data feed is created. When setting up a daily feed, we recommend selecting multiple file delivery, due to the significant performance increases gained when compressing and uncompressing files that are larger than 2 GB. Multiple file delivery makes it easier to process data in parallel. Data files are always split on a complete record and can be easily concatenated after extraction.

One Time Configuration Process

Task	Performed By	Description
Select data columns	Customer	Review the *clickstream data columns* and determine the data you would like to receive. Adobe also provides a recommended column set that can be selected.
Select FTP location	Customer	Select an FTP location where Adobe should deliver data feed files. Adobe can provide FTP hosting for the files if preferred.
Contact Adobe Customer Care to configure the data feed.	Customer	Contact Customer Care through your Supported User and provide: • The report suite that contains the data you want in the feed. • The columns you want in the data set. • Daily or hourly data delivery. If Daily, select single file or multiple file delivery (*multiple* recommended). • FTP hostname, credentials, and path.

Delivery Process

Task	Performed By	Description
Data collection	Adobe	Server calls are collected and processed in Adobe Data Collection servers.
Feed generation	Adobe	After data is processed for the delivery period (previous hour or previous day), the data is exported to the data feed. The feed is stored in delimited format and compressed.
Delivery to customer	Adobe	The compressed data is transferred to customer-hosted or Adobe-hosted FTP site. When complete, a manifest file (or .fin file for older feeds) is transferred indicating that the delivery is complete.

Task	Performed By	Description
Data download	Customer	The customer monitors the FTP for the manifest file. This file contains details on all files that were delivered.
Manifest file processing	Customer	The manifest file is read and each listed file is downloaded.
Data is uncompressed and processed	Customer	Downloaded files are uncompressed and processed.

After you have configured your data feed, continue to *Data Feed Contents* to understand what files you will receive.

Data Feed Contents

This section describes the files found in a data feed delivery.

Manifest File

The manifest file contains the following details about each file that is part of the uploaded data set:

- file name
- file size
- MD5 hash
- number of records contained in the file

The manifest file follows the same format as a Java JAR manifest file.

The manifest file is always delivered last as a separate `.txt` file, so that its existence indicates that the complete data set for that request period has already been delivered. Manifest files are named according to the following:

```
<report_suite_id>_YYYY_MM_DD.txt
```

A typical manifest file contains data similar to the following:

```
Datafeed-Manifest-Version: 1.0
 Lookup-Files: 1
 Data-Files: 1
 Total-Records: 611

 Lookup-File: bugzilla_2012-09-09-lookup_data.tar.gz
 MD5-Digest: af6de42d8b945d4ec1cf28360085308
 File-Size: 63750

 Data-File: 01-bugzilla_2012-09-09.tsv.gz
 MD5-Digest: 9c70bf783cb3d0095a4836904b72c991
 File-Size: 122534
 Record-Count: 611
```

Every manifest file contains a header, indicating the total number of lookup files, data files, and total number of records in all data files. This header is followed by multiple sections containing information for each file included in the data feed delivery.

Some feeds are configured to receive a `rsid_YYYY-MM-DD.fin` file instead of a `.txt` manifest. The `.fin` indicates that the upload is complete, but it contains no metadata about the upload.

Lookup Files

Lookup files do not contain hit data, these are supplemental files that provide the column headers for the hit data, and lookup files to translate the IDs found in the data feed to actual values. For example, a value of "497" in the browser column indicates that the hit came from "Microsoft Internet Explorer 8".

Note that the `column_headers.tsv` and `event_list.tsv` are specific to the data feed and report suite. Other files, such as `browser.tsv`, are generic.

The lookup files are delivered together in a compressed zip named according to the following:

```
<report_suite_id>_<YYYY-mm-dd>-<HHMMSS>-lookup_data.<compression_suffix>
```

- `column_headers.tsv` (customized for this data feed)
- `browser.tsv`
- `browser_type.tsv`
- `color_depth.tsv`
- `connection_type.tsv`

- `country.tsv`
- `javascript_version.tsv`
- `languages.tsv`
- `operating_systems.tsv`
- `plugins.tsv`
- `resolution.tsv`
- `referrer_type.tsv`
- `search_engines.tsv`
- `event_lookup.tsv` (customized for this data feed)

For hourly delivery, lookup files are delivered only with the data for the first hour of each day.

Hit Data Files

Hit data is provided in a `hit_data.tsv` file. The amount of data in this file is determined by the delivery format (hourly or daily, and single or multiple files). This file contains only hit data. The column headers are delivered separately with the lookup files. Each row in this file contains a single server call.

Delivery Contents

The actual files delivered by Adobe vary based on the type of data feed that you have configured. Find the configuration that matches your data feed in the following table for a description of the delivered files.

The time (HHMMSS) indicated in a file name always indicates the beginning of the date range for the data in the file, regardless of when the file was produced or uploaded.

Delivery Format	Description
Daily, single file	After data is collected for a day, you will receive a delivery that contains the following: • a single compressed data file. • A manifest file. The data file is delivered with the following name: `<report_suite>_<YYYY-mm-dd>.<compression_suffix>` Where `<compression_suffix>` is either `tar.gz` or `zip`. When extracted, the data file contains a single `hit_data.tsv` file with all data for that day, as well as the compressed lookup files described above. The hit data file size varies greatly depending on the number of variables actively used and amount of traffic on the report suite. However, on average, a row of data is approximately 500B (compressed) or 2KB (uncompressed). Multiplying this by the number of server calls can provide a rough estimate on how large a data feed file will be.
Daily, multiple file	After data is collected for a day, you will receive a delivery that contains the following: • One or more compressed data files, broken into 2 GB chunks. • A manifest file. Each data file is delivered with the following name: `<index>-<report_suite>_<YYYY-mm-dd>.<compression_suffix>`

Delivery Format	Description
	Where `<index>` is an incrementing file index from *1* to *n*, given *n* files, and `<compression_suffix>` is either `tar.gz` or `zip`. When extracted, each data file contains a single `hit_data.tsv` that contains approximately 2 GB of uncompressed data, as well as the compressed lookup files described above.
Hourly, single file	After data is collected for an hour, you will receive a delivery that contains the following: • a single data file. • A manifest file. The data file is delivered with the following name: `<report_suite>_<YYYY-mm-dd>-<HHMMSS>.<compression_suffix>` Where `<compression_suffix>` is either `tar.gz` or `zip`. When extracted, the data file contains a single `hit_data.tsv` file with all data for that hour. The compressed lookup files described above are delivered only with the data for the first hour of each day.
Hourly, multiple file	After data is collected for an hour, you will receive a delivery that contains the following: • One or more compressed data files, broken into 2 GB chunks. • A manifest file. Each data file is delivered with the following name: `<index>-<report_suite>_<YYYY-mm-dd>-<HHMMSS>.tsv.<compression_suffix>` Where `<index>` is an incrementing file index from *1* to *n*, given *n* files, and `<compression_suffix>` is either `gz` or `zip` When extracted, each data file contains a single `hit_data.tsv` that contains approximately 2 GB of uncompressed data. The compressed lookup files described above are delivered only with the data for the first hour of each day.

Calculating Metrics

Describes how to calculate common metrics using clickstream data feeds.

Pre vs. Post column

Bots

Bots are excluded from data feeds according to the *bot rules* defined for your report suite.

Date filtering

Include rows from the date range you want included by filtering the `date_time` field. The `date_time` field is human readable (for example, `YYYY-MM-DD HH:MM:SS`) and is adjusted to the time zone of the report suite. For example, `date_time starts with "2013-12"` includes hits from December 2013.

Event string

The event string in `event_list` and `post_event_list` contains a comma-delimited list of events, which may have a value and/or a unique ID. We recommend doing all processing on the `post_event_list` because it is de-duplicated and has already applied logic to remove duplicate events with the same ID (see *Event Serialization*).

For event ID to name mapping, see the event lookup delivered with your data feed.

For more information on events, see *Events*.

Formulas for common metrics

The following table contains instructions to calculate several common metrics.

Metric	How to calculate
Page Views	1. Exclude all rows where `exclude_hit > 0`. 2. Exclude all rows with `hit_source = 5,8,9`. These are summary rows uploaded using data sources. See *Hit Source Lookup*. 3. Sum the remaining rows that have `post_page_event = 0`, which indicates a normal page view. See *Page Event Lookup*. You can use similar logic to count custom links: • `post_page_event = 100` to count custom links. • `post_page_event = 101` to count download links. • `post_page_event = 102` to count exit links.
Visits	1. Exclude all rows where `exclude_hit > 0`. 2. Exclude all rows with `hit_source = 5,7,8,9`. 5, 8, and 9 are summary rows uploaded using data sources. 7 represents transaction ID data source uploads that should not be included in visit and visitor counts. See *Hit Source Lookup*. 3. Combine `post_visid_high`, `post_visid_low`, `visit_num`, and `visit_start_time_gmt`*. Count unique number of combinations.

Metric	How to calculate
	*In rare circumstances, internet irregularities, system irregularities, or the use of custom visitor IDs can result in duplicate `visit_num` values for the same visitor ID that are not the same *visit*. To avoid resulting issues, also include `visit_start_time_gmt` when counting visits.
Visitors	1. Exclude all rows where `exclude_hit > 0`. 2. Exclude all rows with `hit_source = 5,7,8,9`. 5, 8, and 9 are summary rows uploaded using data sources. 7 represents transaction ID data source uploads that should not be included in visit and visitor counts. See *Hit Source Lookup* 3. Combine `post_visid_high` with `post_visid_low`. Count unique number of combinations.
Event instances	When an event is set on a hit, `post_event_list` contains the event. The `post_event_list` is de-duplicated and is recommended to determine event instances. For example: `post_event_list = 1,200` Indicates an instance of `purchase` and `event1`. 1. Exclude all rows where `exclude_hit > 0`. 2. Exclude all rows with `hit_source = 5,8,9`. These are summary rows uploaded using data sources. See *Hit Source Lookup*. 3. Count the number of times the event lookup value appears in `post_event_list`.
eVar instances	When an eVar is set on a hit, `event_list` contains an instance of that eVar. For example: `post_event_list = 100,101,106` Indicates an instance of `eVar1`, `eVar2`, and `eVar7`. This means that a value for these three eVars was set on the hit. To calculate instances for eVars, use the same logic explained in *Event instances* above, but count the number of times the eVar lookup appears in the `post_event_list`.

Time Spent	To calculate time spent, you must group hits by visit, then order them according to the hit number within the visit.
	1. Exclude all rows where `exclude_hit > 0`.
	2. Group hits for a visit by concatenating `visid_high`, `visid_low`, and `visit_num`.
	3. Order hits for each visit by `visit_page_num`.
	4. Using *page_event*, filter the types of hits you want.
	5. Find hits where the value you want to track time spent is set. For example:
	`hit 1: post_prop1=red`
	`hit 2: post_prop1=blue`
	6. Subtract the `post_cust_hit_time` for hit 1 from the `post_cust_hit_time` for hit 2 to determine the seconds between these two hits. The result is the time spent for `post_prop1=red`. If this results in a negative number, it indicates that the hit was received out-of-order and the calculation should be discarded.

Metric	How to calculate
	This logic can be extended to calculate time spent for other values. When calculating time spent, Analytics calculates time spent based on the time the value was set in a `track` (`page_event=0`) or `trackLink` (`page_event=10|11|12`) call, to the time of the next page view (`track` call). When reporting time spent for a specific period, marketing reports & analytics and ad hoc analysis evaluate hits beyond the reporting period to determine time spent for values within the reporting period, except when the start and/or end date of the time period is on a monthly boundary. Due to the complexity of these calculations, it might be difficult to match the time spent metrics exactly. Data warehouse does not evaluate hits beyond the reporting period.
Revenue, orders, units	Currency conversion is applied to the `post_product_list` according to the settings for the report suite, so using that column is recommended. 1. Exclude all rows where `exclude_hit > 0`. 2. Exclude all rows with `hit_source = 5,8,9`. 5-9 represent summary rows uploaded using data sources. See *Hit Source Lookup*. 3. Ignore purchase data for rows where `duplicate_purchase = 1`. This flag indicates that the purchase is a duplicate (meaning that a hit with the same `purchaseID` was already recorded). 4. The `post_product_list` uses the same syntax as *s.products*, so you can parse this string to calculate metrics. For example: `;Cross Trainers;1;69.95,;Athletic Socks;10;29.99` By parsing this string, you can determine that 1 pair of cross trainers were purchased for $69.95, and that total revenue from this purchase was $99.94. ***Note:*** *Analytics allows currency events that contain product revenue to be passed in through the events string, so you might need to account for revenue that is not in the products string. See Numeric/Currency Events in s.events.*

Identifying Visitors

Analytics provides several mechanisms by which visitors can be identified (listed in *Analytics Visitor IDs*). Regardless of the method used to identify a visitor, in data feeds the final visitor ID used by Analytics is split across the `post_visid_high` and `post_visid_low` columns, even when using the Marketing Cloud Visitor ID service.

To identify unique visitors:

1. Exclude all rows where `exclude_hit > 0`.
2. Exclude all rows with `hit_source = 5,7,8,9`. 5, 8, and 9 are summary rows uploaded using data sources. 7 represents transaction ID data source uploads that should not be included in visit and visitor counts. See *Hit Source Lookup*
3. Combine `post_visid_high` with `post_visid_low`. All hits across all dates that contain this combination of `post_visid_high` and `post_visid_low` can be considered as coming from same visitor.

If you would like to determine which mechanism was used to determine the visitor ID value (for example, to calculate cookie acceptance), the `post_visid_type` contains a lookup key that indicates which ID method was used. The lookup keys are listed along with the visitor ID mechanisms in the *table below*.

Marketing Cloud Visitor ID

The Marketing Cloud Visitor ID is reported in a separate column, `mcvisid`. Because this ID is reported in its own column, it can be unclear if Analytics is using this ID or a different ID to identify a visitor.

If the marketing Cloud Visitor ID was used to identify the visitor, the ID will be contained in the `post_visid_high` and `post_visid_low` columns and the `post_visid_type` will be set to 5. When calculating metrics, you should use the value from the `post_visid_high` and `post_visid_low` columns since these columns will always contain the final visitor ID.

Analytics Visitor IDs

There are several ways a visitor can be identified in Analytics (listed in the following table in order of preference):

Order Used	Query Parameter (collection method)	post_visid_type column value	Present When
1	*vid (s.visitorID)*	0	s.visitorID is set.
2	*aid (s_vi cookie)*	3	Visitor had an existing s_vi cookie before you deployed the Visitor ID service, or you have a Visitor ID *grace period* configured.
3	*mid (AMCV_ cookie set by Marketing Cloud Visitor ID service)*	5	Visitor's browser accepts cookies (first-party), and the Marketing Cloud Visitor ID service is deployed.
4	*fid (fallback cookie on H.25.3 or newer, or AppMeasurement for JavaScript)*	4	Visitor's browser accepts cookies (first-party).
5	*HTTP Mobile Subscriber header*	2	Device is recognized as a mobile device.
6	*IP Address, User Agent, Gateway IP Address*	1	Visitor's browser does not accept cookies.

Tip: *When using the Adobe Analytics visitor ID as a key for other systems, always use* `post_visid_high` *and* `post_visid_low`*. These fields are the only visitor ID fields guaranteed to have a value with every row in the data feed.*

In many scenarios you might see 2 or 3 different IDs on a call, but Analytics will use the first ID present from that list as the official visitor ID, and split that value across the `post_visid_high` and `post_visid_low` columns. For example, if you are setting a custom visitor ID (included in the "vid" query parameter), that ID will be used before other IDs that might be present on that same hit.

Pre and Post Columns

The pre column contains the data as it was sent to data collection. The post column contains the value after processing.

For example, variable persistence, processing rules, VISTA rules, and currency conversion might change the final value recorded for a variable that appears in the post column. For most calculations you want to use the post column unless you are applying custom business logic (for example, applying a custom formula to determine attribution).

If a column does not contain a pre or a post version (for example, `visit_num`), then the column can be considered a post column. Columns prefixed with "`pre_`" typically contain data that was populated by Adobe and not sent by your implementation. For example, `pre_browser` is populated by Adobe, but `evar1` is populated by your implementation. Both of these columns have a "`post_`" column (`post_browser`, `post_evar1`), which contains the value used by reporting.

Case Sensitivity in Values

Most Analytics variables are considered as case-insensitive for reporting purposes, meaning different case variations are considered to be the same value ("snow", "Snow", "SNOW", and "sNow" are all considered to be the same value). However, for display purposes, case sensitivity is preserved since most customers prefer to be able to send in mixed case characters to display in reports.

When processing the data feed, you can lowercase values for comparison purposes, though you'll likely want to preserve case for display purposes.

If you see different case variations of the same value between the pre and post columns (for example, "snow" in the pre column, and "Snow" in the post column), it indicates that you are passing in both uppercase and lowercase versions of the same value across your site. The case variation in the post column was previously passed in and is stored in the virtual cookie, or was processed around the same time for that report suite. For example:

Hit 1: s.list1="Tabby,Persian,Siamese”;

Hit 2: s.list1=“tabby,persian,siamese”;

When hit 2 is reported in the data feed, the pre column will contain the exact casing passed in (tabby,persian,siamese), but the value from hit 1 is likely persisted for that visit and will be reported in the post column (which will be Tabby,Persian,Siamese) since hit 1 and 2 contain the exact same value when a case-insensitive comparison is performed.

Special Characters

Information about special characters used in the Clickstream data feed.

Special characters in the hit_data file

The following characters have a special meaning in the hit_data file:

Character	Meaning	Description
`\t` (tab character)	End of column	Marks the end of a data field.
`\n` (newline character)	End of row	Marks the end of a data row.
`\` (backslash character)	Escape character	Escapes tab, newline, and backslash when the character was part of the value sent during data collection.

When any of the special characters are preceded by a backslash, they represent a literal character.

Character	Meaning	Description
`\\t`	Tab	Literal tab character. This character was part of the value sent in during data collection.
`\\n`	Newline	Literal newline. This character was part of the value sent in during data collection.
`\\`	Backslash	Literal backslash character. This character was part of the value sent in during data collection.

Special characters in multi-valued variables (events_list, products_list, mvvars)

The following characters have a special meaning in multi-valued variables:

Character	Meaning	Description
`,` (comma character)	End of value	Separates product strings, event IDs, or other values in multi-valued variables.
`;` (semicolon character)	End of sub-value within an individual product value	Separates values associated with an individual product in the `product_list`.
`=` (equals character)	Value assignment	Assigns a value to an event in the `event_list`.

When any of the special characters are preceded by a caret, they represent a literal character.

Character	Meaning	Description
`^,`	Comma	Literal comma character. This character was part of the value sent in during data collection.
`^;`	Semicolon	Literal semicolon character. This character was part of the value sent in during data collection.
`^=`	Equals	Literal equals character. This character was part of the value sent in during data collection.
`^^`	Caret	Literal caret character. This character was part of the value sent in during data collection.

Sample workflow

If some of the columns in your data feed contain user-submitted data, you should check for special characters before separating the data by column or row using `split` or `readLine`, or similar.

Consider the following data:

Browser Width	Browser Height	eVar1	prop1
1680	1050	search\nstring	en
800	600	search\tstring	en

During export, the newline and tab characters in the eVar1 values are escaped. The data feed for these rows appears as follows:

```
1680\t1050\tsearch\\nstring\ten\n
800\t600\tsearch\\tstring\ten\n
```

Calling `readLine()` on the first row returns the following partial string:

```
800\t600\tsearch\
```

Calling `split("\t")` on the second row returns the following string array:

```
800
600
search\
string
en
```

To avoid this, use a solution similar to the following:

1. Starting at the beginning of the file, read until you locate a tab, newline, backslash or caret character.
2. Perform an action based on the special character encountered:
   - Tab - insert the string up that point into a data store cell and continue.
   - Newline - complete the data store row.
   - Backslash - read the next character, insert the appropriate string literal, and continue.
   - Caret - read the next character, insert the appropriate string literal, and continue.

Troubleshooting

Transfer Failures

In the event of an FTP transfer failure (login denied, lost connection, out of quota, etc), Adobe attempts to automatically connect and send the data up to three separate times. If the failures persist, the feed is marked as failed and an email notification is sent.

Resend Options

If you need data for a delivery period resent, contact Customer Care. Adobe stores the generated data feed files for the previous 7 days, so if the request is made within this period data feed files can be resent without being regenerated.

Daylight Savings impact on Hourly Data Feeds

For certain time zones the time will change twice a year due to daylight saving time (DST) definitions. Data feeds honor the time zone for which the report suite is configured. If the time zone for the report suite is one that does not use DST, file delivery will continue normally like any other day. If the time zone of the report suite is one that does use DST, file delivery will be altered for the hour in which the time change occurs (usually 2:00 am).

When making STD -> DST time transitions (“Spring Forward”), the customer will only get 23 files. The hour that is skipped in the DST transition is simply omitted. For example, if the transition occurs at 2 AM, they’ll get a file for the 1:00 hour, and will get a file for the 3:00 hour. There will be no 2:00 file, since at 2:00 STD, it becomes 3:00 DST.

When making DST -> STD transitions, (“Fall Back”), the customer will get 24 files. However, the hour of transition will actually include 2 hours’ worth of data. For example, if the transition occurs at 2:00 am, the file for 1:00 will be delayed by one hour, but will contain data for two hours. It will contain data from 1:00 DST to 2:00 STD (which would have been 3:00 DST). The next file will begin at 2:00 STD.

No Data for a Time Period

You can optionally configure a data feed to deliver a manifest file if no data is collected for a specific period. If you enable this option, you'll receive a manifest file similar to the following:

```
Datafeed-Manifest-Version: 1.0
 Lookup-Files: 0
 Data-Files: 0
 Total-Records: 0
```

Data Processing Overview

Before processing hourly or daily data, data feeds waits until all the hits that entered data collection within the timeframe (day or hour) have been written out to data warehouse. After that occurs, data feeds collects the data with timestamps that fall within the timeframe, compresses it, and sends it via FTP. For hourly feeds, files are typically written out to data warehouse within 15-30 min after the hour, but there is no set time period. If there was no data with timestamps that fall within the timeframe, then the process tries again the next timeframe. The current data feed process uses the `date_time` field to determine which hits belong to the hour. This field is based on the time zone of the report suite.

Contact and Legal Information

Information to help you contact Adobe and to understand the legal issues concerning your use of this product and documentation.

Help & Technical Support

The Adobe Marketing Cloud Customer Care team is here to assist you and provides a number of mechanisms by which they can be engaged:

- *Check the Marketing Cloud help pages for advice, tips, and FAQs*
- *Ask us a quick question on Twitter @AdobeMktgCare*
- *Log an incident in our customer portal*
- *Contact the Customer Care team directly*
- *Check availability and status of Marketing Cloud Solutions*

Service, Capability & Billing

Dependent on your solution configuration, some options described in this documentation might not be available to you. As each account is unique, please refer to your contract for pricing, due dates, terms, and conditions. If you would like to add to or otherwise change your service level, or if you have questions regarding your current service, please contact your Account Manager.

Feedback

We welcome any suggestions or feedback regarding this solution. Enhancement ideas and suggestions for Adobe Analytics *can be added to our Customer Idea Exchange*.

Legal