

Handbuch zur Vorlage für Google Tag Manager

Der “*Conversions Manager*”

Für zentrale Event- & Conversion-Verwaltung



2024 Markus Baersch

Inhalt

Was ist der “Conversions Manager”?	2
Ein Tag für “alles”? Der Conversions Manager	3
Tealium in GTM?.....	3
Conversions Manager installieren.....	5
Datenschicht-Format für Ereignisse/Conversions definieren.....	8
Konfiguration von (Ereignissen und) Conversions.....	8
Ereignisdaten in Datenschicht.....	9
Eigenschaften eines Ereignisses.....	10
Conversion-Regeln.....	13
Sorry, leider nerdig: Steuerung via Präfix.....	13
Einfache Suche nach Treffern in URL.....	14
RegEx-Suche in URL.....	15
Suche nach vollständiger Übereinstimmung.....	15
“RegEx Light”: Mehrere Vergleichswerte nutzen.....	16
Auslösung per Ereignisnamen im dataLayer.....	16
Einsatz des “Custom rule input”.....	18
Custom value in Regeln nutzen.....	20
Nerdmodus 1000: Extraktion aus Custom Input.....	21
Kombinierte Bedingungen?.....	23
Das “Custom” Feld für alle anderen Parameter.....	23
Werte als dataLayer Variablen nutzen.....	25
Variablen in Tags nutzen.....	27
Auslösen des Conversions Manager Tags.....	29
Ereignisbasierte Regeln erfordern ereignisbasierte Trigger.....	29
Trigger für nutzende Event / Conversion Tags.....	30
Conversions Manager Toolbox.....	34
Regeln verwalten.....	35
Zugriff freigeben.....	36
Regeln bearbeiten.....	37
Tabellenspalten statt JSON!.....	38
Tag aktualisieren.....	39
Variablen und Trigger automatisiert erstellen.....	39
Beispielrechnung zu Einsparungen.....	42
Danke!	44

Was ist der “Conversions Manager”?

Der Conversions Manager ist eine benutzerdefinierte Vorlage für den Google Tag Manager, der eine alternative Methode anbietet, um Ereignisse und Conversions an einer zentralen Stelle zu verwalten.

Dieses Handbuch beschreibt die Rahmenbedingungen und den Einsatz der Vorlage im Google Tag Manager. Ebenso wird eine speziell für den Conversions Manager erstellte Toolbox beschrieben, mit deren Hilfe die Pflege von Ereignis- und Conversiondaten deutlich leichter fällt.

Die Beschreibung in diesem Handbuch ist ein (wesentlicher) Auszug aus einem E-Book über Effizientes Tag Management für den Google Tag Manager. Es ist derzeit in der Entstehung. Unter

<https://www.markus-baersch.de/download-index.html>

wird es nach Fertigstellung einen passenden Link geben. Auch danach aber soll diese Anleitung weiter bestehen, um den Einstieg und die Verwendung der Conversions Manager Vorlage zu erleichtern.

Wer Kommentare, Vorschläge oder Fehlerkorrekturen hat, sendet diese bitte an mail@markus-baersch.de.

Damit: Happy Tagging!

Ein Tag für “alles”? Der Conversions Manager

Der “Conversions Manager” ist ein Tag, das einem ganz bestimmten Zweck dient: Der größtmöglichen Konsolidierung von Tags für einen einzelnen Dienst in Google Tag Manager.

Ursprünglich “nur” für Conversions gedacht, können damit für beliebig viele Empfänger (wie Google Analytics & Ads, Facebook etc.) alle zur Messung eines **Ereignisses** oder einer Conversion erforderlichen Angaben an einer **zentralen Stelle** verwaltet werden - einschließlich der **Regeln**, nach denen diese ausgelöst werden sollen.

Wenngleich die Bedienung und Nutzung nicht intuitiv ist und einige Anleitung erfordert, lohnt sich der Einsatz vor allem dort, wo jenseits von Basis-Tracking und E-Commerce viele weitere Ereignisse vermessen werden, die nicht nur an einen einzelnen Empfänger wie GA4, sondern auch weitere Dienste gesendet werden sollen.

Hinweis: Ob sich der Einsatz des Conversions Managers lohnt oder nicht, muss anhand der Einsparungen bestimmt werden, die damit erzielt werden können. Eine Beispielrechnung findet sich am Ende dieser Beschreibung, die so lang ausfällt, weil der Conversions Manager **viel Flexibilität auf kleinstem Raum** bieten muss. Das Ergebnis ist oft erklärungsbedürftig - so auch hier.

Wer sich ein Bild machen will, ohne dieses Kapitel komplett durchzuarbeiten, kann sich vor dem Eintauchen in die Details dieser Anleitung bei YouTube unter https://youtu.be/zdK_mqrgBEg vorab über die allgemeine Nutzung informieren.

Wie bei der in vielen Containern bereits umgesetzte Vereinheitlichung des Trackings für E-Commerce soll durch Einsatz dieses Tags ein dataLayer entstehen, der alle erforderlichen Daten beinhaltet, um damit die jeweiligen Tags zu betreiben, welche Empfängerdienste zur Messung von Ereignissen anbieten.

Tealium in GTM?

Im Extremfall wird darüber sogar das E-Commerce Tracking abgebildet, so dass jenseits der Basis-Trackingcodes (Google-Tag, Conversion Linker, Facebook Pageview etc.) für jeden Empfänger nur noch ein einziges Tag benötigt wird, das die ganze Arbeit übernehmen kann.

Dieser Ansatz ist nicht einzigartig. In Tealium wird z. B. üblicherweise nur ein Tag je Empfänger verbaut und alle erforderlichen Parameter werden dynamisch mit Werten bestückt. Aus der Datenschicht, Cookies, Extensions (im weitesten Sinne benutzerdefinierte JavaScript Variablen) und anderen Quellen, die genau wie beim GTM nutzbar sind.

Dieser Ansatz kann im Google Tag Manager auf unterschiedliche Arten umgesetzt werden, auch ohne den Conversions Manager. Weil aber bei Nutzung von Such- und RegEx-Tabellen wiederum Pflege an mehreren Stellen erfolgen muss, basieren viele Ansätze auf JavaScript Variablen, die mehr oder weniger verschachtelte Sammlungen von Eingangs- und Ausgangswerten verwalten. Ein Beispiel:

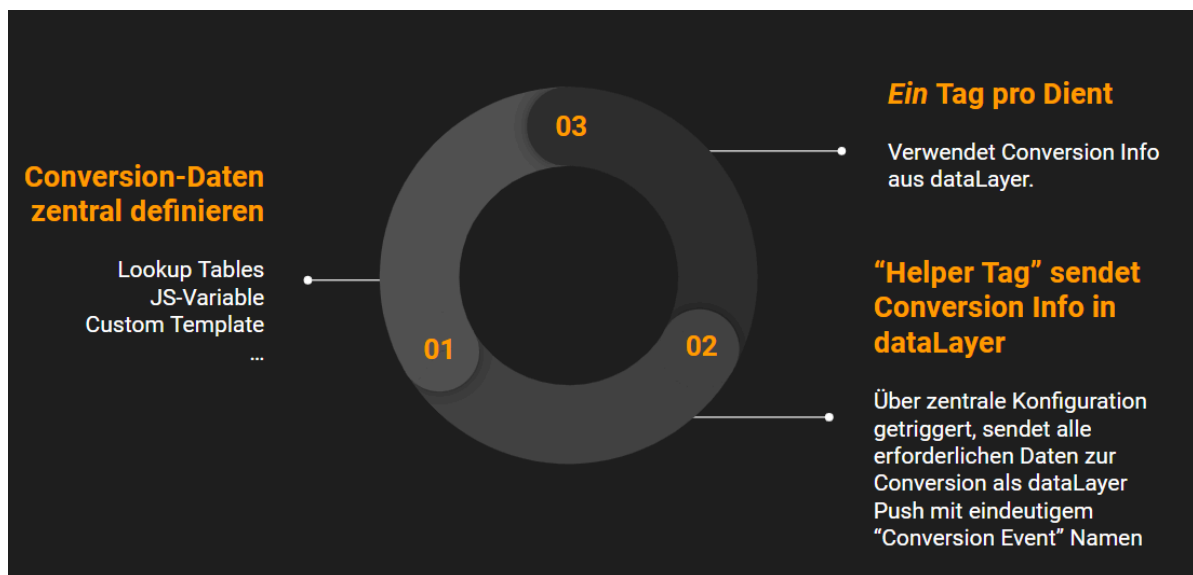


```
Custom JavaScript ⓘ  
1 function() {  
2  
3   var conversionData = [  
4  
5     {  
6       pattern: "/thank-you/double",  
7       name: "Training Fall 2024",  
8       value: 0,  
9       ads_label: "LRXTConNkckDEKqF7sMD",  
10      facebook: true  
11    },  
12    {  
13      pattern: "/newsletter/confirm/",  
14      name: "Newsletter DOI",  
15      value: 0,  
16      ads_label: "g4T-CJv_-fYCEKqF7sMD",  
17      facebook: true  
18    },  
19  ]  
20 }
```

Diese Daten werden dann von einem Tag (wiederum ein HTML Tag mit JavaScript) genutzt, um in der Datenschicht alle erforderlichen Angaben zu hinterlegen, damit die entsprechenden Tags gefüttert und ausgelöst werden können. Das Prinzip hat also drei Stufen:

1. Die (möglichst zentrale) **Verwaltung** der Conversion-Daten
2. Das Auslösen von **dataLayer Pushes** bei Ereignissen und Conversions mit den jeweiligen Daten wie Pixel IDs, Ereignisnamen, Conversion-Wert, Google Ads Conversion Label und anderen

3. **Ein Tag je Empfänger**, das mit den Informationen aus der Datenschicht versehen und ausgelöst wird

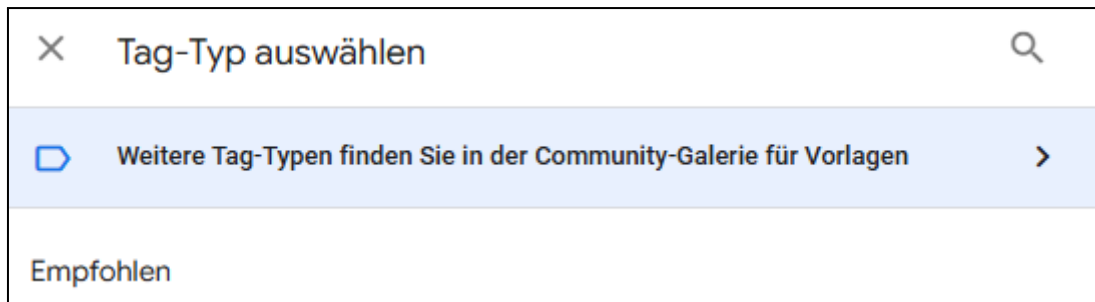


Obschon JavaScript für individuelle Lösungen durchaus nützlich ist, ist das Auslösen der dataLayer Events und die Konfiguration der Daten dabei immer verteilt, denn mindestens ein Tag und eine Variable sind im Spiel. Ansonsten wird ein *sehr* umfangreiches HTML Tag betrieben, in dem alle Änderungen manuell eingetragen werden müssen... mit dem Risiko, dabei den Code fehlerbehaftet zu hinterlassen, nur weil irgendwo ein Komma fehlt oder Formatvorgaben missachtet wurden und somit das komplette darüber gesteuerte Tracking ausfällt.

Um dies zu vermeiden, bietet der Conversions Manager alle benötigten Einstellungen direkt in der Tag-Oberfläche an und füttert den dataLayer mit Ereignis-/Conversion-Daten.

Conversions Manager installieren


Die Conversions Manager Tag-Vorlage kann direkt im GTM bei der Neuanlage eines Tags über den Link zur Community-Galerie und Suche nach "*Conversions Manager*" installiert werden.





Damit erhält man eine zentrale Verwaltungsoberfläche für (fast) beliebige Ereignisse oder Conversions und deren Eigenschaften, was nicht bedeutet, dass es nicht Anwendungsfälle gibt, bei denen man mehr als ein Tag dieses Typs im Container betreibt. Im Abschnitt über das Auslösen des Conversions Manager Tags wird diese Option näher erläutert.

Nach der Installation sieht ein noch nicht individuell bestücktes Tag ohne Conversion-Regeln und -Daten wie abgebildet aus, wobei zum Zweck der Beschreibung lediglich der Optionsschalter zur Aktivierung des “*Custom input*” aktiviert wurde:

Tag-Typ


 **Conversions Manager**
mbaersch




 **Tag-Berechtigungen**

3 Berechtigungen >

Event name ?



Empty value placeholder ?



▼ Conversion data


Add at least one conversion rule and add conversion data to be pushed to the dataLayer along with the defined conversion event name. NOTE: you can use a Google Spreadsheet to edit this table. Find a link at <https://github.com/mbaersch/conversions-manager/#readme>.

Conversion rule	Event name	Value	Conversion label	Custom data	Aktionen
<div>Zeile hinzufügen</div>					

▼ Advanced Options

☒ Enable custom rule input ?

Custom value ?



Die folgenden Abschnitte beschreiben im Detail, wozu die angebotenen Optionen dienen. Vor der Beschreibung der Bestückung des Regelwerks für die Auslösung von Ereignissen werden zunächst die weiteren Elemente betrachtet, um ein Gesamtbild zu zeigen. Vorab nur so viel:

1. **Regeln** bestimmen anhand der aktuellen URL, dem Datenschicht-Ereignis oder anderen Merkmalen, ob ein Ereignis ausgelöst werden soll
2. Bei Übereinstimmung wird ein eigenes **Datenschicht-Ereignis** ausgelöst, bei dem alle Angaben zu Eigenschaften aus dem Conversions Manager beigefügt werden

3. **Tags** für einen oder mehrere Dienste nutzen die Informationen aus der Datenschicht zum **Senden** des Ereignisses / der Conversion, indem diese durch das Datenschicht-Ereignis getriggert werden

Damit entspricht die Funktionsweise genau dem oben beschriebenen Prinzip.

Datenschicht-Format für Ereignisse/Conversions definieren

Den Anfang macht ein Ereignisname, der genutzt wird, wenn das Tag Informationen über ein darüber verwaltetes Ereignis in den dataLayer schreibt. Denn das ist alles, was der Conversions Manager macht: Wenn er ausgelöst wird, wird die Tabelle der definierten Regeln durchgegangen. Das Tag löst für jeden Treffer je eine Conversion aus. Es kann also sein, dass mehrere - gleiche oder verschiedene-Regeln mit dem aktuellen Zustand übereinstimmen, so dass Daten für mehrere Conversions / Ereignisse in die Datenschicht geschrieben werden.

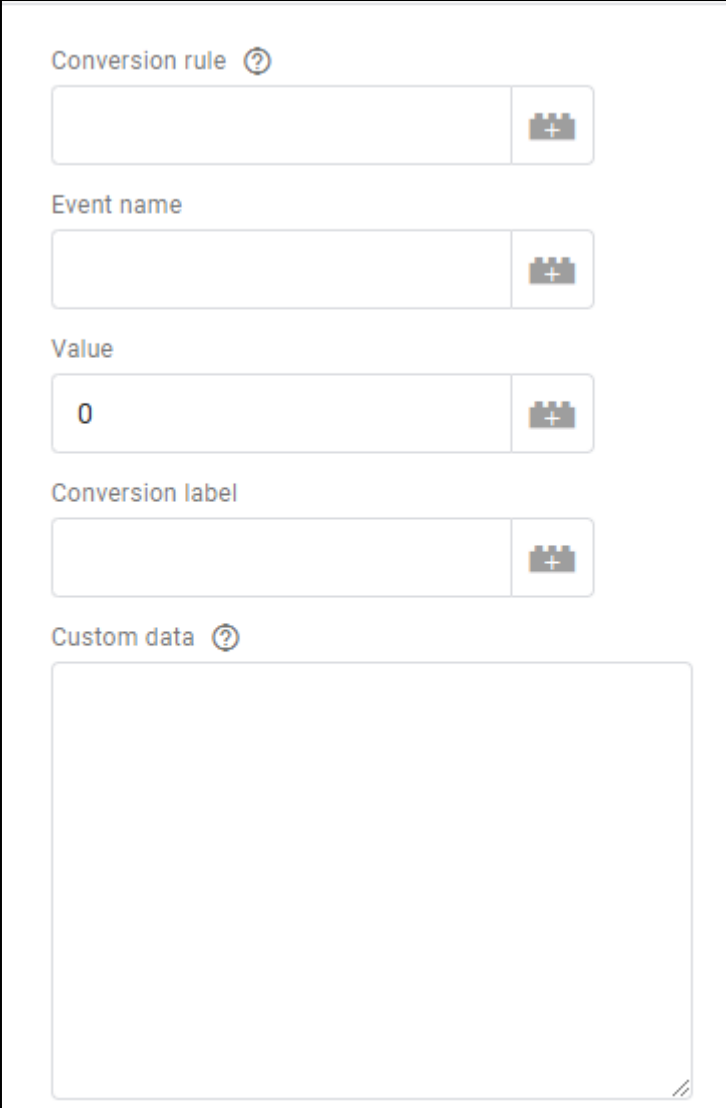
Mit "Zustand" ist hier die zum Zeitpunkt des Feuerns des Conversions Manager Tags geltenden Bedingungen gemeint, also z. B. die derzeitige URL, das zuletzt stattgefundene dataLayer Event, Variablenwerte etc.

Das Datenschicht-Ereignis bekommt den **Namen**, welcher im ersten Eingabefeld "*Event name*" angegeben wurde. Es ist mit "*managedConversionEvent*" vorbelegt. Findet dieses Ereignis im dataLayer statt, werden damit dann im dritten Schritt gemäß der obigen Liste die Tags der nutzenden Dienste ausgelöst.

Das nächste Eingabefeld dient zur Definition eines **Standardwertes** für alle nicht belegten Eigenschaften, wenn diese in die Datenschicht geschrieben werden. Dazu mehr bei der folgenden Beschreibung der Tabelle mit den Regeln und deren Nutzung.

Konfiguration von (Ereignissen und) Conversions

Im Zentrum steht die Tabelle "*Conversion data*", in der in je einer Zeile Informationen zu Conversions - oder anderen beliebigen Ereignissen - abgelegt werden. Dabei müssen nicht immer alle Felder ausgefüllt werden. Es genügt für eine sinnvolle Nutzung eine Regel ("*Conversion rule*") und mindestens ein mit Eigenschaften des Ereignisses bzw. der Conversion gefülltes Feld. Die Neuanlage und Bearbeitung geschieht in einem eigenen Dialogfenster der Tag Manager Oberfläche.



The screenshot shows a configuration form for a conversion rule. It contains five input fields, each with a small icon of a box with a plus sign to its right. The fields are labeled: 'Conversion rule' (with a help icon), 'Event name', 'Value' (containing the number '0'), 'Conversion label', and 'Custom data' (with a help icon). The 'Custom data' field is a larger text area.

Die hier gemachten Angaben zu einem Ereignis, das dann stattfindet, wenn die definierte Regel greift, werden bei Ausführung des Tags in die Datenschicht geschrieben. Fehlende Daten werden mit dem ausgewählten Platzhalter ersetzt.

Ereignisdaten in Datenschicht

Ein solcher Push in den dataLayer kann so aussehen:

```
dataLayer.push({  
  event: "managedConversionEvent",  
  conversionInfo: {  
    name: "NONE",  
    value: 42,  
    ads_label: "ACcDefG2345hIjKL-mN",
```

```
    custom: {send_to_pinterest: false, fb_event_name: "Lead",
lead_type: "Contact"}
  },
  gtm.uniqueEventId: 295
})
```

Neben dem Namen des dataLayer Events sieht man in einem eigenen *conversionInfo* Block alle Angaben zum aktuellen Ereignis. Der Standardwert *"NONE"* bei der fehlenden *"Name"*-Eigenschaft dient dazu, zuverlässige Trigger für alle Tags zu bauen, die das Feld verwenden (dazu später mehr).

Eigenschaften eines Ereignisses

Im Conversions Manager werden die Parameter aus dem oben abgebildeten verwaltet:

1. Auslöse-Regel
2. Ereignisname
3. Wert
4. Ads Label
5. Custom

Wie oben zu sehen ist, landen die Angaben 2-5 in der Datenschicht, wenn die unter 1 definierte Auslöse-Regel greift und der Conversions Manager ausgeführt wird. Was ebenfalls zu sehen ist: Der Inhalt des *custom* Schlüssels ist im Beispiel keine einfache Zeichenkette oder Zahl, sondern eine verschachtelte Struktur.

Einer von zwei Gründen dafür ist sowohl einfach wie traurig: GTM selbst warnt im Vorlagen-Editor davor, mehr als fünf Spalten zu nutzen.

Spalte hinzufügen

Achtung: Parametertabellen mit mehr als fünf Spalten werden in der GTM-Benutzeroberfläche möglicherweise nicht korrekt dargestellt

Und statt im UI eine scrollbare Tabelle anzuzeigen, ist die Darstellung bei Überschreiten dieser Grenze tatsächlich nicht sehr ansehnlich (aber theoretisch immer noch nutzbar):

The screenshot shows the 'Conversion rule' configuration interface. It contains four input fields, each with a plus icon to its right, and a 'Custom data' section at the bottom.

- Conversion rule**: /contact/success
- Event name**: generate_lead
- Conversion value**: {{value by customer type}}
- Conversion label**: MMcBhf6272Gaf3244-kBh4
- Custom data**:

```
{  
  "send_to_pinterest": true,  
  "fb_event_name": "Lead",  
  "lead_type": "Contact",  
}
```

Damit würde auf einer Kontakt-Danke-Seite, deren URL in der Regel als Suchmuster definiert ist...

- ein Ereignisname "*generate_lead*" (geeignet z. B. für GA4) mit
- einem dynamisch generierten Wert (mittels der eingesetzten Variable),
- einem Label für eine Google Ads Conversion
- und Steuerungsdaten für weitere Dienste

in den dataLayer geschrieben. Das Ergebnis sieht also ganz ähnlich aus wie im Beispiel oben.

Die Nutzung dieser Daten anhand von Variablen im Tag Manager geschieht dann durch die Ereignis-/Conversion-Tags für GA4, Google Ads und die weiteren Empfänger. Der Vorgang wird später in einem eigenen Abschnitt noch im Detail

erklärt. Zunächst aber zu den Regeln, die als erste Eingabe einer Zeile als *Conversion rule* in der Tabelle erforderlich sind.

Conversion-Regeln

Regeln sind - wie schon beschrieben - das Pendant eines Triggers, der nur dann auslöst, wenn bestimmte Bedingungen erfüllt sind. Mit diesen Regeln bestimmt das Tag, ob und welche Ereignisse inkl. ihrer Daten in die Datenschicht geschrieben werden sollen. Wann immer das Tag ausgeführt wird, werden **alle Regeln nacheinander abgearbeitet** und jeder Treffer führt zu einem Conversion Event Push im dataLayer.

Dabei können die folgenden Optionen zur Definition einer Regel verwendet werden:

- Suche nach teilweiser Übereinstimmung ("*contains*")
- Suche nach vollständiger Übereinstimmung ("*equals*")
- Suche mit regulären Ausdrücken ("*matches RegEx*")

Durchsucht werden dabei wahlweise:

- Die aktuelle URL
- Der Name des aktuellen dataLayer Events
- Beliebige weitere Variablen via "*Custom rule input*"

Sorry, leider nerdig: Steuerung via Präfix

Mit welchem Verfahren an welcher Stelle gesucht wird, wird über eine **besondere Syntax von Präfixen** gesteuert, die einem Suchbegriff bzw. RegEx-Suchmuster vorangestellt werden. Es können aber auch einfache Regeln definiert werden, ohne sich mit dieser eigenen "Grammatik" des Conversions Manager Regelwerks auszukennen.

Ein Tipp: Fast jedes Eingabefeld bietet einen Hilfetext an, sodass bei der Anlage neuer Regeln nicht immer diese Anleitung zur Hand genommen werden muss.

Conversion rule ?

re:\ueber-uns\

Event name

generate_lead

Value

20

Conversion label

4adstest4

Custom data ?

Use partial match (default): Enter a string that has to be contained in the URL (including parameters) for creating a dataLayer event. You can split multiple patterns with a "|" when using this default mode without any prefix. Note: You can use the prefix "uc:" for "URL contains" if you want all rules to use the same structure. RegEx: To use regular expressions with URL, use "re:" or "ur:" as prefix (example: "re:something_.*\.html\$") Equals: Use "ue:" to define a pattern that has to completely match the current URL. To compare with events name instead of URL: Use prefix "ec:" to compare with the current event name ("event contains"), "ee:" for "event equals" or "er:" for "event matches regex".

Zudem bietet die Conversions Manager Toolbox dabei eine weitere Hilfestellung (siehe unten).

Einfache Suche nach Treffern in URL

Um eine *Conversion Rule* einzutragen, genügt es im einfachsten Fall, den Teil einer URL, eines Pfades oder gesuchte Parameterbezeichnung inkl. Wert einzutragen. Das Standardverfahren, das bei einer Eingabe ohne ein Präfix zum Einsatz kommt, ist die Suche nach einer teilweisen Übereinstimmung **in der vollständigen URL**, so dass Host, Pfad, Dateiname und alle Parameter genutzt werden können.

Sinnvolle Beispiele für eine Regel nach diesem Verfahren:

- `/contact/success/` für die Suche nach einer Übereinstimmung im Pfad an beliebiger Stelle
- `example.com/contact/success/` für die Suche nach einer Übereinstimmung im Pfad, aber nur auf der genannten Domain
- `/contact/?success=1` für die Suche nach einer Übereinstimmung im Pfad inkl. Parameter und Wert
- `success=1` für die Suche nach einem Parameter inkl. Wert, unabhängig von Position, Pfad und Hostnamen

Wie man jetzt schon merkt, ist gerade das letzte Beispiel nicht ideal, denn jeder Parameter, der mit *success* endet und dessen Wert mit *1* beginnt, würde zu

einer Auslösung führen. Wer es genauer haben möchte, würde im Fall eines normalen Triggers zu einem regulären Ausdruck greifen.

RegEx-Suche in URL

Das geht auch hier, allerdings ist durch den begrenzten Platz in der Tabelle keine Möglichkeit, weitere Felder bereitzustellen. Deshalb kann man mit einem Präfix dafür sorgen, dass der Conversions Manager eine Regel nicht nach dem Standardverfahren auswertet, sondern die Eingabe als RegEx betrachtet und die URL damit nach einer Übereinstimmung sucht.

Für eine RegEx-Suche in der URL dient ein dem Ausdruck vorangestelltes **“re:”**. Alles, was nach dem Doppelpunkt folgt, ist der reguläre Ausdruck. Auch dazu einige Beispiele:

- `re:\Vsuccess\.html$` für die Suche irgendwo im Pfad, der mit dem angegebenen Dateinamen enden muss und keine Parameter haben darf (\$ = “Ende”)
- `re:(shop|www)\.example\.com\contact\success\` Suche nach exaktem Pfad nach Übereinstimmung, aber nur auf den Subdomains www oder shop
- `re:\Vcontact\.*[?]&success=1(&|$)` Suche nach einer Übereinstimmung im Pfad inkl. Parameter und Wert an beliebiger Stelle, Wert muss exakt passen
- `re:[?]&success=1(&|$)` wie oben, aber nur für den exakten Wert und beliebige Position des Parameters

Dieses Verfahren kann man auch einsetzen, um nach vollständigen Übereinstimmungen zu suchen, also dem Pendant zu *“ist gleich”* bei einer Trigger-Regel. Das ist allerdings mühselig durch die ganze Arbeit zur Maskierung von Sonderzeichen wie Slashes und Punkten, die in URLs nun mal häufig vorkommen, nur um einen vollständigen Treffer zu definieren.

Suche nach vollständiger Übereinstimmung

Um RegEx zu vermeiden, nur um komplette Suchbegriffe verwenden zu können, gibt es ein weiteres Präfix für den Suchbegriff. Dieses lautet **“ue:”**. Das Kürzel steht für **“url equals”**.

Dabei ist zu beachten, dass es *vollständige* URLs sind, mit denen verglichen wird, also inkl. Protokoll, Host, Domain und so weiter. Während dies bei URLs nicht immer sinnvoll ist, wird es in den weiteren Optionen zur Bildung einer Regel (wie dem dataLayer Ereignisnamen, siehe unten) deutlich sinnvoller erscheinen.

Um beim Pfad aus dem zweiten RegEx-Beispiel zu bleiben, kann statt des dort verwendeten Ausdrucks auch folgende Syntax genutzt werden, um eine einfachere Regel für den Shop zu erstellen:

`ue:https://shop.example.com/contact/success/`

Da die andere Variante ebenso nur dann auslösen sollte, wenn die URL keine weiteren Pfade, Dateinamen oder Parameter nach dem `/success/` enthält, ist diese Regel gleichwertig, aber viel kürzer und besser zu lesen. Allerdings müsste diese Zeile komplett wiederholt werden, um auch für den `www`-Host eine Auslösung zu erreichen, die bei der RegEx-Version ebenfalls abgedeckt war. Dazu dient ein Trick, mit dem mehrere Vergleichswerte angegeben werden können.

“RegEx Light”: Mehrere Vergleichswerte nutzen

Statt einer Wiederholung der gleichen Regel können in einer Regel mehrere Werte angegeben werden. Das funktioniert gleichermaßen, wenn nach vollständiger oder teilweiser Übereinstimmung gesucht werden soll. Dazu werden mehrere Werte durch ein Pipe-Zeichen “|” getrennt, wie es auch in einer RegEx “Aufzählung” der Fall wäre. Daher stammt auch der Spitzname dieser Option. Ein Beispiel:

- `/contact/success|callback/success/` dient dazu, nach teilweiser Übereinstimmung im Pfad zu suchen und sowohl bei `/contact/success/` als auch `/callback/success/` in der URL auszulösen

Dieser Trenner kann auch verwendet werden, wenn nach einer vollständigen Übereinstimmung gesucht werden soll, führt dann allerdings zu sehr langen und unübersichtlichen Regeln, so dass RegEx dazu vermutlich die bessere Wahl ist.

Da man aber nicht nur in URLs suchen kann, sondern auch in `dataLayer` Ereignisnamen und über das “*Custom input*” in beliebigen anderen Vergleichswerten, ist diese Funktion dort deutlich nützlicher.

Auslösung per Ereignisnamen im dataLayer

Statt in der URL zu suchen, kann der Conversions Manager auch bei bestimmten Datenschicht-Events getriggert werden, die in Trigger-Regeln genutzt werden sollen. Näheres dazu folgt im Abschnitt zum Auslösen des Conversions Manager Tags.

Die gebräuchlichste Form dabei ist die vollständige Übereinstimmung. Das Präfix baut sich genau wie bei der URL zusammen, nur lautet es hier **“ee:”**. Statt des **“u”** für URL wird also ein **“e”** für *Event* verwendet (**“event equals”**).

Damit auch nach teilweiser Übereinstimmung gesucht werden kann, existiert das weitere Präfix **“ec:”** für **“event contains”**. Und auch eine RegEx-Variante ist mittels **“er:”** möglich (**“event matches RegEx”**). Dazu wieder Beispiele:

- **ee:purchase** für das GA4 Standardereignis beim E-Commerce Kaufabschluss, komplette Übereinstimmung
- **ec:view_item** für alle Events, die den Suchbegriff *view_item* enthalten. Würde also auch bei *view_item_list* auslösen. Soll dies nicht der Fall sein, auch hier ee statt ec verwenden.
- **ee:add_to_cart|remove_from_cart** würde für beide dataLayer Events ausgelöst, die in der “|” getrennten Liste stehen. Hier daher viel sinnvoller als bei URLs zu nutzen
- **er:^(add_to|remove_from)_cart\$** macht das gleiche, aber als RegEx

Über ein *“Custom input”* kommen weitere Präfixe hinzu, mit denen man bei der Erstellung von Regeln arbeiten kann, um auch jenseits von URLs und Ereignissen arbeiten zu können, also z. B. mit Click-Elementen etc.

Weil dieses ganze Regelwerk zwar logisch aufgebaut ist, aber immer wieder erfordern würde, die passenden Präfixe nachzuschlagen oder aus den Hilfetexten der Oberfläche zu holen, existiert eine Erstellungshilfe in der Toolbox, welche im Detail weiter unten beschrieben wird. Damit können Vergleichswerte, Übereinstimmungstypen und Suchbegriffe in einer Tabelle eingetragen werden, was deutlich einfacher ist und mehr an die Erstellung von Trigger-Bedingungen erinnert. Hier eine Abbildung als Vorschau (und zur Beruhigung ;)):

Source	Match Type	Value	Result
Page URL	Contains	/demo/?param=1	/demo/?param=1
Page URL	Equals	https://domain.com/demo/?param=1	ue:https://domain.com/demo/?param=1
Page URL	Matches RegEx	\test\,*\?foo=bar	re:\test\,*\?foo=bar
Event Name	Contains	view_item	ec:view_item
Event Name	Equals	view_item	ee:view_item
Event Name	Matches RegEx	view_+	er:view_+
Custom	Contains	foo	cc:foo
Custom	Equals	foo	ce:foo
Custom	Matches RegEx	foo.+	cr:foo.+
Page URL	Contains	<add here>	<add here>

Die damit erstellten Regeln aus der Spalte “*Result*” kann man in einem weiteren Arbeitsblatt der Toolbox einfügen, auf dem die Regeln der “*Conversion data*” Tabelle ebenfalls komplett extern gepflegt werden können.

Ein Hinweis schon jetzt: Um nur die Werte eines Formelergebnisses einzufügen, dient in Google Sheets die Tastenkombination **CTRL+SHIFT+V**.

Zurück zum Tag und den dortigen Optionen, die weitere Regeln ermöglichen:


Einsatz des “Custom rule input”

Für alle Vergleichswerte außer URLs und Ereignissen in dataLayer kann ein weiteres Eingabefeld “*Custom value*” aktiviert werden, in dem eine oder mehrere beliebige Variablen hinterlegt werden können, um darin ebenfalls nach teilweisen oder kompletten Übereinstimmungen bzw. per RegEx zu suchen.

Advanced Options

☒ Enable custom rule input ?

Custom value ?



Das abgebildete Beispiel erzeugt z. B. eine Eingabe, in der man gezielt nach CSS IDs und / oder Klassen des angeklickten Elementes suchen kann, indem beiden Teilen eine eindeutige Markierung vorangestellt wird, die weder in der einen noch anderen Variable vorkommen kann. Da es um CSS Eigenschaften geht, ist hier die passende CSS Notation verwendet worden.

Der selbst gebildete Vergleichswert muss aber nicht zwingend so aussehen wie abgebildet. Wer nur mit der `{{Click ID}}` arbeiten will oder einer anderen Variable, kann diese auch als einzige Eingabe hier unterbringen, ohne sich Gedanken über Trennzeichen machen zu müssen. Diese "eine Variable" wiederum kann zudem einen beliebig komplex aus anderen Variablen zusammengestellten Wert bereitstellen; je nach Anforderungen an Trigger für Conversions und Events.

Für mehr Vergleichswerte können diese z. B. mit einem beliebigen, immer *gleichen* Trennzeichen versehen werden und man sucht dann im gesamten Inhalt nach Übereinstimmungen. *Irgendein* Trennzeichen empfiehlt sich allerdings immer, denn so kann man zumindest den Anfang eines Variablenwertes identifizieren, was das Erstellen des Regelmusters vereinfacht. Ein Beispiel:

```
{{Click ID}}|{{Click Classes}}|{{Click URL}}|{{Click  
Text}}|{{Referrer}}
```

Wer einzelne Einträge referenzieren will, kann im *Custom value* für jedes Element einen eindeutigen Schlüssel hinterlegen. Mit einem "i." für die ID, "c." für die Klasse etc. kann man i. d. R. alles kennzeichnen, solange in Variablenwerten keine Doppelpunkte vorkommen - oder zumindest keine, die mit einem der verwendeten Zeichen zusammenfallen. Daher kann man bei Bedarf auch ein anderes - oder mehrere - Trennzeichen verwenden. Das kann z. B. ein "#i:" oder "##i:." sein, um damit zuverlässige Suchbegriffe oder RegEx Muster für den richtigen Wert erstellen zu können.

Solch eine Auflistung kann so aussehen:

```
i:{{Click ID}} c:{{Click Classes}} u:{{Click URL}} t:{{Click  
Text}} r:{{Referrer}}
```

Hierbei ist noch Referrer als Beispiel für beliebige weitere Variablen hinzugenommen worden, die in Regeln genutzt werden können, wenn sie Teil des *Custom value* sind. Die gleiche Liste kann man auch als valides JSON formulieren:

```
{"i":"{{Click ID}}","c":"{{Click Classes}}","u":"{{Click  
URL}}", "t":"{{Click Text}}","r":"{{Referrer}}"} 
```

Das würde zwar die Suchbegriffe wieder etwas verlängern, hat aber einen anderen Vorteil, der im übernächsten Abschnitt vorgestellt wird.

Custom value in Regeln nutzen

Um einen selbst kombinierten Wert für Regeln zu nutzen, wird das Präfix mit einem “c” für “*custom*” gebildet. Dementsprechend lauten die voranzustellenden Zeichenketten hierfür:

- “**cc:**” für “***custom contains***”
- “**ce:**” für “***custom equals***”
- “**cr:**” für “***custom matches RegEx***”

Genau wie bei URLs und Ereignissen sind damit Regeln zu erstellen, die allerdings auf die jeweiligen Inhalte des *Custom value* angewendet werden. Wer als Eingabe in Regeln z. B. ausschließlich auf Eigenschaften eines Klick Elements zugreifen möchte, kann einen “*Custom value*” nach dem Muster verwenden:

```
#{{Click ID}}.{{Click Classes}}:{{Click URL}},{{Click Text}}
```

Ein exemplarischer Wert, der sich z. B. bei Klick auf einen Button ergeben könnte, wäre z. B. diese Zeichenkette:

```
#myButtonId.myClass:https://www.example.com/somelink.html,Cli  
ck here to go to some link!
```

Damit ließen sich z. B. folgende Regeln erstellen:

- **cc:#myButtonId.** für Klicks auf eine Schaltfläche, deren CSS ID “*myButtonId*” lautet. Die Abgrenzung durch den Punkt im Muster sorgt dafür, dass die ID genau übereinstimmen muss und nicht nur mit der gewünschten Zeichenkette beginnt (sonst würde eine ID wie “*myButtonId1*” ebenfalls passen)
- **cc:.myClass:** für Klicks auf eine Schaltfläche, deren CSS Klasse “*myClass*” lautet. Dies wird aber nur funktionieren, wenn es keine weiteren Klassen gibt - für alles andere muss es dann RegEx sein, um im Bereich zwischen Punkt und Doppelpunkt gezielt nach einem Klassennamen zu suchen
- **cc:#myButtonId.MyClass** für Klicks auf eine Schaltfläche, deren CSS ID “*myButtonId*” und die Klasse (genau) “*myClass*” lautet

- `cc:#myButtonId.MyClass` für Klicks auf eine Schaltfläche, deren CSS ID `"myButtonId"` und die Klasse (genau) `"myClass"` lautet
- `cr:Click here` RegEx für Klicks auf einen Text, der mit `"Click here"` beginnt
- `cr:https.+example\.com.*`, RegEx zur Suche nach `"example.com"` mit beliebigen Hostnamen in der geklickten URL

Ein Vergleich mit `"ce:"` für komplette Übereinstimmungen ist hingegen nur bei einzelnen Eingangswerten sinnvoll, nicht aber bei einem Input, der aus so vielen Einzelwerten besteht.

Dennoch kann man mit `"equals"` auch bei komplexeren `"Custom"` Werten arbeiten. Dazu ist aber eine weitere Variante der Regel-Syntax zuständig: Die Extraktion aus einem JavaScript Objekt bzw. einem JSON String. Da lacht das Nerd-Herz.

Nerdmodus 1000: Extraktion aus Custom Input

Hier die Wiederholung eines der Beispiele aus dem vorherigen Abschnitt. Es zeigt einen JSON String, der aus einer Menge Klammern, Anführungszeichen und Variablen zusammengesetzt war. Die Abfrage über `"ce:"` und selbst via RegEx wäre darin ziemlich kompliziert:

```
{"i":"{{Click ID}}","c":"{{Click Classes}}","u":"{{Click URL}}","t":"{{Click Text}}","r":"{{Referrer}}"}
```

Viel praktischer wäre es, wenn man über die Schlüssel auf die jeweiligen Werte wie den Referrer zugreifen könnte.

Noch praktischer wäre ein echtes JavaScript Objekt, das man per JS Variable oder Template (ja, auch dafür gibt es Vorlagen!) zusammenstellen könnte. Darin wären dann auch nicht mehr die sehr kurzen Schlüsselnamen wie `"i"` und `"r"` etc. erforderlich, welche nur eingesetzt wurden, um das Eingabefeld überschaubar zu halten.

Eine solche benutzerdefinierte JavaScript Variable, die eine ganze Liste verschiedener Vergleichswerte enthält, könnte z. B. so aussehen:

```
function() {  
  return {  
    pageHostname: "{{Page Hostname}}",  
    pageReferrer: "{{Referrer}}",  
  }  
}
```

```
clickElement: "{{Click Element}}",
clickId: "{{Click ID}}",
clickClasses: "{{Click Classes}}",
clickUrl: "{{Click URL}}",
clickText: "{{Click Text}}",
clickTarget: "{{Click Target}}",

formElement: "{{Form Element}}",
formId: "{{Form ID}}",
formClasses: "{{Form Classes}}",
formUrl: "{{Form URL}}",
formTarget: "{{Form Text}}",

ecItemString: "{{ecommerce.items}}",
ecValue: "{{ecommerce.value}}",
ecTransactionId: "{{ecommerce.transaction_id}}",

//add whatever variables might be relevant for your
trigger rules
}
}
```

Hieraus nun einzelne Werte zu extrahieren, indem man die ganze riesige Zeichenkette nutzt und mit “ce:” oder “cr:” darin herumzusuchen, ist nicht nur mühselig, sondern bietet reichlich Raum für Fehltreffer.

Daher ist ein **weiterer Präfix-Typ “p”** nutzbar, der zwar auch den *Custom value* durchsucht, aber bei dem **ein einzelner Wert anhand des Schlüssels extrahiert** und zum Vergleich genutzt wird. Alles andere aus der Variable oder dem JSON String wird ignoriert. Das erlaubt die gezielte Auswertung aller hier vorhandenen Daten.

Wie bei den anderen Typen kann also mit

- “**pe:**” nach vollständiger Übereinstimmung
- “**pc:**” nach Teiltreffern
- “**pr:**” mittels RegEx

gesucht werden. Nach dem ersten Doppelpunkt folgt hierbei aber zunächst der **Name des Feldes**, in dem gesucht werden soll und erst nach einem weiteren Doppelpunkt der Suchbegriff oder das RegEx Muster. Anhand der oben gezeigten JS Variable könnten also z. B. folgende Regeln erstellt werden:

- `pc:pageTitle:404` sucht nach "404" im Seitentitel, um bei Fehlerseiten ausgelöst zu werden
- `pc:clickText:Click here` Alternative Fassung für Klicks auf einen Text, der mit "Click here" beginnt, ohne mit RegEx
- `pr:clickUrl:https.+example\..com` Ebenfalls das Pendant zum Beispiel oben bei normaler Nutzung des Custom value per "cr:" zum Ermitteln externer Klicks für `example.com`

Damit das funktioniert, muss der Eingabewert über *Custom value* unbedingt entweder bereits ein JavaScript Objekt oder ein gültiger JSON String sein, damit das Tag die Werte extrahieren kann. Auch ist zu beachten, dass nur eine Ebene möglich ist. Wer also z. B. "ecommerce.items" auslesen will, muss das als Zeichenkette tun oder die Eigenschaften des ersten Produkts separat in sein Vergleichsobjekt übernehmen.

Kombinierte Bedingungen?

Die einzige Einschränkung, die auch alle anderen Regeln betrifft: Es kann stets nur eine Bedingung definiert werden. Kombinationen von Bedingungen sind zwar theoretisch umsetzbar, aber schon jetzt ist die Definition komplex genug. Wer also mehr braucht als das, was hier möglich ist, muss die entsprechenden Tags entweder ohne Hilfe des Conversions Managers verwalten oder mit dem passenden Trigger ein Hilfs-Tag feuern, welches wiederum ein eindeutiges Ereignis in den dataLayer schreibt. Dieses wäre dann wiederum im Conversions Manager mit "ee:eindeutiger_eventname" als Regel erfassbar.

Das "Custom" Feld für alle anderen Parameter

Ist die passende Regel gefunden, gilt es in den weiteren Feldern die Eigenschaften der Conversion oder des Events zu definieren, welche in die Datenschicht geschrieben werden sollen.

Ein **Name** kann z. B. für GA4-Ereignisse dienen oder als *Custom Event Name* für das Facebook Pixel... oder beides. Meint: Wie die verfügbaren Felder zur Definition eines Namens, **Werts** und **Labels** genutzt werden (können) hängt ganz vom Anwendungsfall ab. So kann auch der Name des GA4-Ereignisses, wenn denn eines ausgelöst werden soll, auch immer gleich sein und die Information aus dem

“*name*” des `dataLayer Pushes` wird dann als Ereignisparameter genutzt. Alle Einsatzmöglichkeiten sind denkbar.

Wo diese drei Angaben nicht ausreichen und auch der weitere Wert, der in das letzte Feld unter “*Custom*” eingetragen wird, allein nicht genügt, um alle geplanten Verbraucher (GA4, Google Ads, Facebook etc.) mit den erforderlichen dynamischen Werten für Parameter etc. zu versorgen, kann man im *Custom* Feld auch mehr unterbringen.

Um eine theoretisch unbegrenzte Anzahl von Eigenschaften zu verwalten, bietet das Tag im Feld *Custom* deshalb zunächst einmal deutlich mehr Platz zur Eingabe als die anderen Felder.

Während man hier Daten für mehrere Dienste ablegen kann, indem man eine Liste erstellt, bei der an einer bestimmten Position ein bestimmtes Merkmal steht (ähnlich wie den Beispielen zum *Custom value* bei den Regeln), ist in diesem Fall der Zugriff und die Nutzung etwas sperrig, denn man müsste den kombinierten Wert zuerst aus dem `dataLayer` lesen und danach zerlegen, um den passenden Einzelwert zu erhalten.

Die Werte selbst können / sollten wiederum beliebige Formen annehmen können. Hier einige Beispiele:

- ein separater Ereignisname für Facebook
- wiederum ein ein anderer für Pinterest
- vom Standard abweichende IDs, wenn mehrere Konten eines Dienstes bedient werden (wie z. B. mehrere Google Ads Konten)
- zusätzliche Angaben, die in Ereignisparameter oder Nutzereigenschaften eines GA4 Events bestimmen sollen
- Kategorie, Aktion etc. für Ereignisse in Piwik PRO oder Matomo
- was auch immer die nutzenden Dienste an variablen Daten benötigen, um Conversions oder andere Events zu verarbeiten, die vom Conversions Manager ausgelöst werden

Um diese Liste unbekannter - und nicht immer gleich gestalteter - Felder zu verwalten, ist **JSON** die beste Wahl. Um ein einfaches **Beispiel** zu machen, sollen in diesem Feld drei Werte verwaltet werden:

1. Ein logischer Schalter (Werte sind also *true* oder *false*) namens ***send_to_facebook***, mit dem gesteuert wird, ob das aktuelle Ereignis auch an Facebook gesendet werden soll.
2. Ein beliebiger String als ***matomo_category***. Wenn diese mit einem Wert besetzt ist, soll ein Matomo Event ausgelöst werden, das diese Kategorie und den Namen des Ereignisses aus dem Feld "*name*" nutzt
3. eine ***google_ads_id***, die nur bei einigen Conversions verwendet werden soll (also nicht bei allen Regeln vorhanden ist)

Um diese Werte in der *Custom* Spalte für eine Zeile einzutragen, müsste folgende Eingabe erfolgen, um die entsprechenden Werte zu bestimmen:

```
{ "send_to_facebook":true,  
  "matomo_category":"Click Conversions",  
  "Google_ads_id":"1234567890"  
}
```

Die Formatierung der Eingabe ist beliebig, aber auf die ganzen Anführungszeichen bei den Schlüsselnamen kann man leider nicht verzichten, damit es ein gültiges JSON ist (siehe <https://www.json.org/json-en.html>).

Ein weiteres Problem: Wenn einmal mehrere Regeln nach diesem Muster generiert wurden und dann später ein neuer Dienst wie Pinterest hinzukommt, so dass ein neuer Schlüssel (z. B. "*send_to_pinterest*":*true/false*) hinzu kommt, muss dieser manuell bei allen entsprechenden Zeilen hinzugefügt werden, bei denen eine Einstellung vorgenommen werden muss.

Klingt viel zu kompliziert? Nicht die Flinte ins Korn werfen! In der weiter unten beschriebenen **Toolbox** wird das ganze Thema auf die Pflege einer ganz normalen Tabelle in Google Sheets reduziert. Es ist daher **keine manuelle Eingabe und Pflege von JSON** erforderlich!

Um die Funktionsbeschreibung abzuschließen, fehlt noch die Verwendung der Daten, die der Conversions Manager erzeugt.

Werte als dataLayer Variablen nutzen

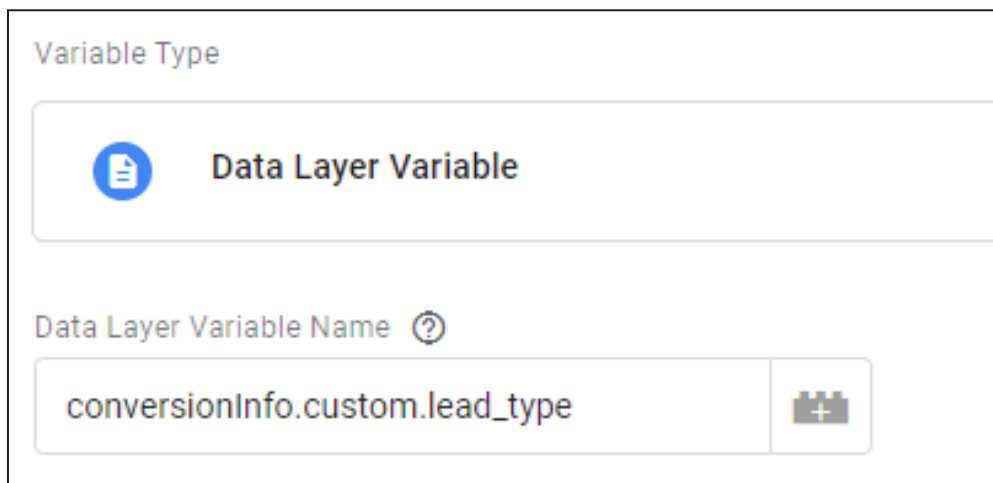
Weil man vermutlich nicht stets *alle* Angaben für alle Conversions benötigt, bietet es sich an, im JSON Objekt (nennen wir es der Einfachheit halber so) nur die vom **Standard abweichenden Einstellungen** zu verwalten. Das bedeutet, dass man

keinen separaten “Schalter” für einen Dienst (wie Facebook Pixel) benötigt, wenn dieser mit den anderen bereits vorhandenen Daten veraltet werden kann. Wenn z. B. alle Events, die auch an GA4 gehen, ebenfalls an Facebook gesendet werden, verwendet man den Ereignisnamen aus dem ersten Eingabefeld der *Conversion data* Zeile und benötigt keinen eigenen Eintrag.

Sollen aber die meisten Events auch an Facebook gehen, kann eine Variable auf einen Schlüssel mit einem Namen wie z. B. *send_to_facebook* zugreifen, der nur bei einigen Conversions über das Custom Feld mit *false* belegt in den *dataLayer* gelangt und ansonsten ein *true* als Standardwert.

Im umgekehrten Fall kann die Vorgabe dieser Variable *false* zurückliefern und nur bei ausgewählten Conversions / Ereignissen im Conversions Manager mit einem “*send_to_facebook*”: *false* ausgeschlossen werden.

Denn der Zweck hinter allen Feldern ist stets die Verwendung der jeweiligen Werte in einer Datenschicht-Variable, wo sie aus der vom Conversions Manager hinterlegten Struktur entnommen und dann in Tags verwendet werden. Sendet man z. B. einen *lead_type* als Feld innerhalb von *Custom*, um diesen dann als Parameter für GA4 Events zu nutzen, sieht die passende Variable so aus:




The screenshot shows a configuration window for a 'Variable Type'. Under the 'Variable Type' header, there is a button labeled 'Data Layer Variable' with a document icon. Below this, the 'Data Layer Variable Name' field is shown with a help icon. The text 'conversionInfo.custom.lead_type' is entered into the field, and a plus icon is visible to the right of the input box.


Hier gibt es keinen Standardwert, weil bei Fehlen einer solchen Angabe (vermutlich) kein Wert zugewiesen werden soll.

Bei anderen Feldern, die vielleicht nicht immer für jede Conversion aus dem *custom* Bereich entnommen werden können, ist ein entsprechender Standardwert sinnvoll. Die folgende Abbildung zeigt eine Variable für die Google Ads Conversion ID. Diese wird bei einzelnen Conversions definiert, da diese in ein abweichendes


Konto gesendet werden sollen. Alle anderen Conversions gehen (bei Fehlen des Schlüssels) an die ID, die als Standardwert definiert wurde.

Variable Type

 **Data Layer Variable**


Data Layer Variable Name 

conversionInfo.custom.ads_id




Data Layer Version


Version 2



☒ Set Default Value

Default Value 

947618474





Variablen in Tags nutzen


Mit diesen Variablen werden dann schlussendlich die Tags bestückt, welche die Daten an die jeweiligen Empfänger senden. GA4 Ereignisname, Ereignisparameter und alles andere kommt dabei also aus dem dataLayer, wo es der Conversions Manager abgelegt hat. Am Beispiel einer Google Ads Conversion werden drei Variablen verwendet, die die ID, das Label und den Wert dynamisch übergeben.


Tag Configuration

Tag Type

**Google Ads Conversion Tracking**
Google Ads

 **Conversion Linker tag found in container.**

Conversion ID 
{{Google Ads | Conversion ID}}

Conversion Label 
{{Conversion Label}}

Conversion Value
{{Conversion Value}}

Currency Code
EUR

Advanced Settings

Tag firing options
Once per page

- Die Variable `{{Google Ads Conversion ID}}` wird dabei nach dem oben gezeigten Vorbild mit einem Vorgabewert belegt und bedarfsweise aus dem dataLayer mit einer anderen ID versehen - oder ist einfach eine Konstante, wenn nur ein Konto im Einsatz ist.
- Das `{{Conversion Label}}` und der `{{Conversion Value}}` kommen ebenso als dataLayer Variablen aus den Infos des Conversions Managers. Ausgelöst wird dieses Tag dann, wenn das im Conversions Manager definierte Ereignis (Standard ist *managedConversionEvent*) im dataLayer ausgelöst wird. Die ganzen Angaben aus der Tabelle für diese Conversion sind wie oben gezeigt im Pfad *conversionInfo* zu finden.

Aber wann sollte das Conversions Manager Tag genau ausgelöst werden?

Auslösen des Conversions Manager Tags

Damit der Conversions Manager die Regeln abarbeiten und alle Daten für die nutzenden Tags in die Datenschicht schreiben kann, muss er immer dann ausgelöst werden, wenn eine der Regeln greifen kann.

Basieren alle Regeln ausschließlich auf Seitenaufrufen, wäre also ein einzelner Trigger ausreichend, der auf allen Seiten feuert. Dazu den eingebauten Trigger *“Alle Seiten”* zu verwenden, ist allerdings nicht immer die ideale Lösung. Denn wenn vom Conversions Manager ein oder mehrere Einträge zu Conversions in den dataLayer geschrieben werden, soll der zugehörige eindeutige Datenschicht-Ereignisname als Trigger für die nutzenden Conversion bzw. Ereignis-Tags dienen. Wenn diese aber **auf Zustimmung angewiesen** sind - was meistens der Fall sein sollte- und / oder gewartet werden muss, bis der Consent Mode initialisiert oder andere Rahmenbedingungen geklärt sind, landen die Conversion-Daten zu früh im dataLayer.

Daher ist für alle URL-basierten Regeln der beste Zeitpunkt zum Feuern des Tags, wenn diese ganzen Dinge geklärt sind. Das ist üblicherweise dann der Fall, wenn das Consent Management Tool die aktuelle Zustimmungslage kundgetan hat. Bei Usercentrics gibt es dazu das *“consent_status”* Event. Andere Systeme arbeiten ähnlich und schreiben die Zustimmungslage mit einem oder mehreren Ereignissen in die Datenschicht. Da dieses Problem im bestehenden Container bereits für andere Tags gelöst ist, sollte die Bestimmung einfach sein.

Ereignisbasierte Regeln erfordern ereignisbasierte Trigger


Wenn es zudem Conversion-Regeln gibt, die auf Ereignissen basieren, muss das Tag entweder bei *allen* Ereignissen feuern (was selten sinnvoll ist) oder die entsprechenden Trigger für die relevanten Ereignisse werden alle dem Conversions Manager zugewiesen.

Genauso verhält es sich bei Klicks, abgesendeten Formularen oder anderen Dingen, die man über den *Custom value* für Conversion-Regeln nutzen mag. Nur selten lohnt es sich also, einen Trigger für *alle* Klicks anzulegen und zu nutzen, wenn nur wenige Klicks per Regel im Conversions Manager zu Ereignissen führen. Denn bei jedem Aufruf des Tags müssen alle Regeln ausgewertet werden.


Ein Kompromiss kann so aussehen, wenn viele verschiedene Klicks, aber nur zwei E-Commerce-Ereignisse (hier sind es *add_to_cart* und *purchase*) und eine Menge Seitenaufrufe anhand der URL für die Regeln des Conversions Manager relevant sind:

Trigger


Auslösende Trigger +

 **Click - All Elements**
Alle Elemente −


ODER

 **consent_status**
Benutzerdefiniertes Ereignis −

ODER

 **EEC | add_to_cart**
Benutzerdefiniertes Ereignis −

ODER

 **EEC | purchase**
Benutzerdefiniertes Ereignis −

Sollten hingegen nur wenige Klicks wichtig sein, sind individuelle Klick Trigger die bessere Wahl - oder nur ein einzelner, der lediglich bei den entsprechenden Klick-Elementen, URLs oder sonstwie eingeschränkt ausgelöst wird.

Hinweis: Wenn Regeln für Ereignisse und URLs mit solchen gemischt werden, welche auf CSS Klassen oder IDs von geklickten Elementen oder Formularen oder ähnlichen Daten bestehen, ist Vorsicht geboten. Denn wenn der Conversions Manager bei jedem Seitenaufruf *und* Ereignissen oder Klicks ausgelöst wird, kann es zu unerwünschten Effekten kommen.

Denn *jede* übereinstimmende Regel führt beim Ausführen des Conversions Managers zu einem dataLayer-Push. - jedes Mal. Wenn ein Klick-Trigger das Tag erneut auf einer Seite auslöst, auf der die URL bereits zu einer Conversion führt, wird der zweite Durchlauf schließlich die Klick-Conversions auslösen und zusätzlich die bereits getrackte Conversion aus der passenden URL-Regel *erneut*. Um dieses Problem zu vermeiden, kann es sinnvoll sein, verschiedene Conversions Manager Tags für verschiedene Regeltypen zu erstellen und separat mit Triggern zu versehen.


Trigger für nutzende Event / Conversion Tags

Bei den Tags für GA4, Google Ads, Facebook etc. gibt es grundsätzlich zwei verschiedene Möglichkeiten:

1. **Separate kombinierte Trigger für jeden Dienst:** Auslösen der Tags für Dienst X (z. B. GA4) beim Event des Conversions Manager (das per Vorgabe "*managedConversionEvent*" heißt), wenn die benötigte Variablen für den Betrieb des Tags (für G4 z. B. der Name unter *conversionInfo.name*) vorhanden ist
2. **Blockieren:** Auslösen aller Tags mit einem gemeinsamen Trigger für das Conversions Manager Event ("*managedConversionEvent*") und per Ausnahme für jeden Dienst dafür sorgen, dass dies nur dann passiert, wenn das erforderliche Feld mit einem Wert belegt ist.

Für den ersten Fall würde ein einzelner Trigger für alle Dienste so aussehen, welche auf das angegebene "*name*" Feld angewiesen sind:

Triggertyp

 Benutzerdefiniertes Ereignis

Ereignisname

managedConversionEvent

Diesen Trigger auslösen bei:


conversionInfo.name ist nicht gleich NONE


Hier kommt der Standardwert unter "*Empty value placeholder*" aus dem Tag zum Einsatz, mit dem über die oben abgebildete Regel sichergestellt werden kann, dass bei Fehlen des Wertes keine Auslösung erfolgt.

Dieser Trigger wird dem GA4 Tag und / oder anderen zugeordnet, die nichts als den Namen benötigen. Ein eigener Ausnahmetrigger ist in diesem Fall nicht erforderlich. Es existiert aber ggf. dennoch bereits eine Zuordnung zu einer Ausnahme, die dafür sorgt, dass ein Tag bei fehlender Zustimmung nicht feuert. Solche blockierenden Trigger sind hier dementsprechend nach wie vor zu verwenden.


Trigger


Auslösende Trigger

 **managedConversionEvent + name**
Benutzerdefiniertes Ereignis



Ausnahmen

 **Blocking | No Analytics**
Benutzerdefiniertes Ereignis




Benötigt man zwingend mehr als einen Variablenwert aus dem Conversions Manager, um ein Ereignis sinnvoll für einen Dienst zu vermessen, sind Ausnahmen nach dem zweiten beschriebenen Muster "*Blockieren*" möglicherweise sinnvoller.

Obschon man mehrere Prüfungen in einem Trigger mit kombinierten Bedingungen packen kann, kann die Arbeit mit Ausschlüssen flexibler sein. Je nachdem, wie viele Dienste welche Daten - also wie viele gemeinsame Kombinationen einzelner Werte aus dem Conversions Manager - benötigen. Arbeitet das bisherige Trigger-Konzept bereits mit Ausnahmen, kann auch bei weniger komplexen Kombinationen weiter mit blockierenden Triggern gearbeitet werden, wenn dies auch vorher schon die bevorzugte Variante war.

Der passende blockierende Trigger bei fehlen eines Namens im dataLayer würde so aussehen:

Triggertyp

 Benutzerdefiniertes Ereignis

Ereignisname

☒ Übereinstimmung mit regulärem Ausdruck verwenden

Diesen Trigger auslösen bei:


☐ Alle benutzerdefinierten Ereignisse ☒ Einige benutzerdefinierte Ereignisse

Diesen Trigger auslösen, wenn ein Ereignis eintritt und all diese Bedingungen erfüllt sind


Bei Tags sieht die Liste der zugeordneten Trigger dann dementsprechend etwas anders aus. Ausgelöst wird mit dem allgemeinen Trigger für das Conversion-Event und dann blockiert, wenn die erforderlichen Daten *nicht* vorhanden sind.

Trigger


Auslösende Trigger

 **managedConversionEvent**
Benutzerdefiniertes Ereignis

Ausnahmen

 **Blocking | No Analytics**
Benutzerdefiniertes Ereignis

ODER

 **Blocking | No Event Name**
Benutzerdefiniertes Ereignis

Braucht ein Tag mehr als nur den Namen, können weitere blockierende Trigger hinzugefügt werden.

Da man bei einer umfangreichen Nutzung des Conversions Manager eine ganze Menge Variablen und Trigger benötigt, liefert die Conversions Manager Toolbox

Hilfestellung. Hierüber können sowohl Variablen für (neue) Felder angelegt werden, als auch kombinierte Trigger nach dem ersten Prinzip für die Auslösung.

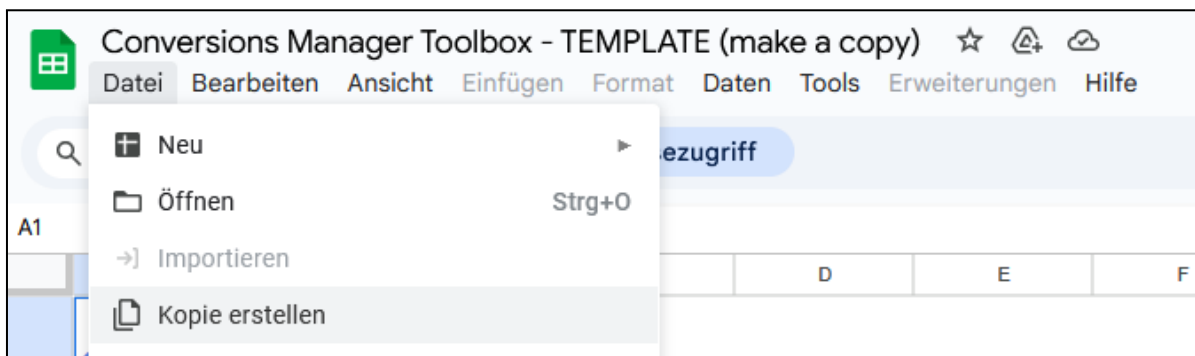
Conversions Manager Toolbox

Da die Anlage und Pflege von Conversion-/Ereignisdaten in der Tabelle des Tags dank des begrenzten Platzes mühselig ist, gibt es mit der Toolbox eine Alternative.

Dabei werden bestehende Regeln aus dem Tag (wenn vorhanden) in ein Google Spreadsheet über die API eingelesen. Sie können dort bequem in einem Arbeitsblatt bearbeitet und ergänzt werden. Anschließend wird das Tag mit den neuen Einstellungen versehen - ebenso via API.

Um die Toolbox zu nutzen, wird eine Kopie einer Vorlage benötigt. Dazu das Google Spreadsheet unter dem folgenden Link öffnen und eine Kopie anfertigen:

https://docs.google.com/spreadsheets/d/1k1L3Cludh2c8fyX3jnYrGqCrCqCvFmrx7E_bGzPBtts



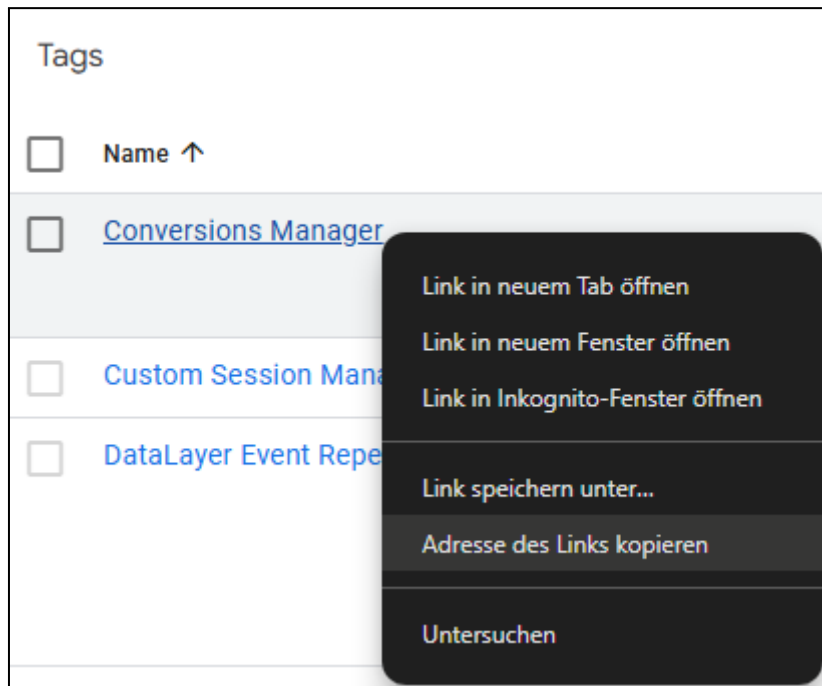
Das erste Arbeitsblatt liefert neben einer kurzen Anleitung in Zelle C2 ein Eingabefeld für den Pfad zum Conversions Manager Tag, das bearbeitet werden soll.

A screenshot of the 'Conversions Manager Toolbox' form. The title bar says 'Conversions Manager Toolbox' and 'Version: 1.0.6.1 | 2024/11/01'. Below the title bar, there's a text input field labeled '"Conversions Manager" Tag Link:' with the placeholder text '<Paste link to existing Conversions Manager tag from GTM here or use "Browse" button to select from container>'. To the right of the input field is a blue 'Browse' button. Below the input field, there's a hint: 'Hint: remember to update the link after creating a new version / workspace! :)'. On the left side, there are three lines of text: 'Last read: 30.10.2024 02:44:06', 'Last update / write: 30.10.2024 06:34:04', and 'Last variable update: 30.10.2024 02:19:15'.

Dieser Pfad wird im folgenden Format angegeben:

<https://tagmanager.google.com/container/accounts/xxx/containers/yyy/workspaces/zzz/tags/123>

Den Link erhält man, wenn man im GTM in der Liste der Tags den Link kopiert, mit dem man per Klick auf den Namen eines Tags die Bearbeitung öffnet. Dazu das Kontextmenü mit einem Rechtsklick öffnen und die entsprechende Option wählen, die bei Chrome *“Adresse des Links kopieren”* heißt.

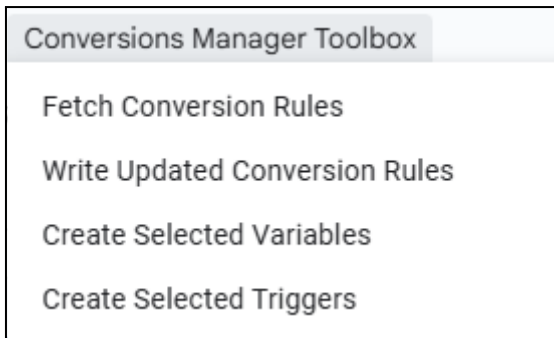


Dieser Link wird auf der ersten Seite in Zelle C2 eingetragen.

Der Pfad zum Tag kann alternativ über den *Browse*-Button herausgesucht werden, was allerdings bei mehreren Konten / Containern etwas mühselig ist. Nicht wundern: In diesem Fall wird kein kompletter Link in die Zelle eingetragen, sondern nur der relevante Teil, welcher ab *accounts* beginnt. Am obigen Beispiel wäre der Pfad also nur *accounts/xxx/containers/yyy/workspaces/zzz/tags/123*.

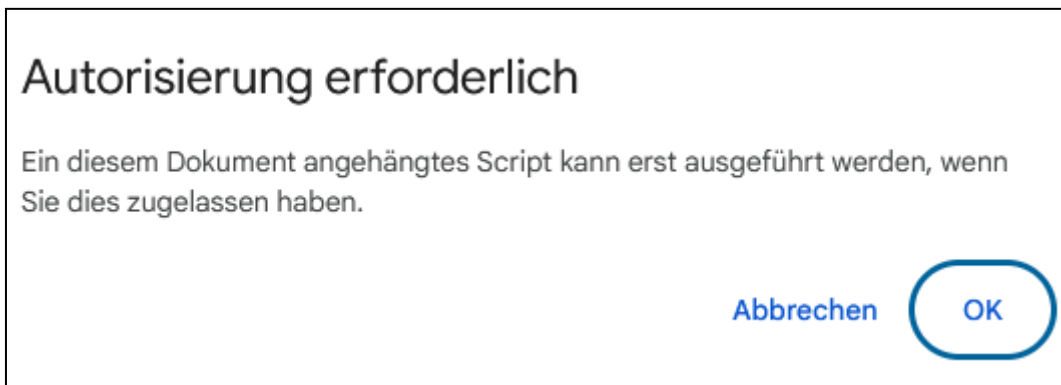
Regeln verwalten

Ist der Pfad definiert, können bestehende Regeln über das Menü *“Conversions Manager Toolbox”* zunächst mittels *“Fetch Conversion Rules”* eingelesen werden.



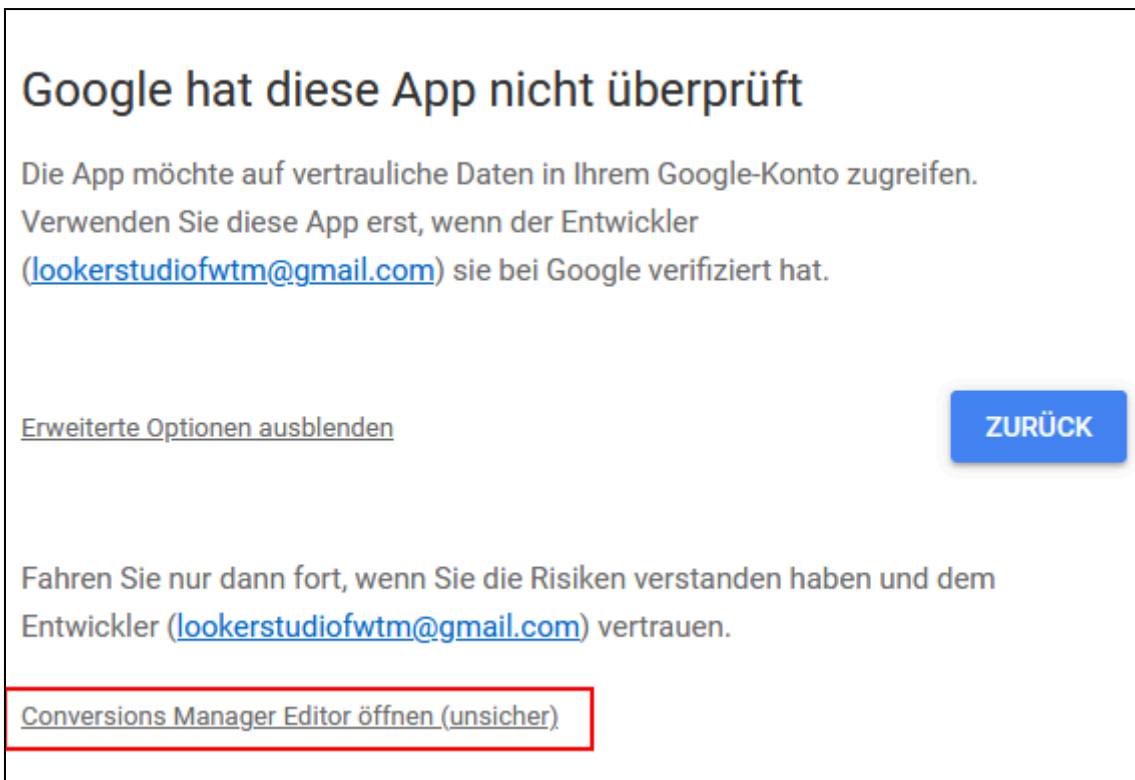
Zugriff freigeben

Bei der ersten Verwendung einer der Funktionen aus dem Menü wird eine Autorisierung erforderlich.



Da das Apps Script der Toolbox nicht an ein Projekt gebunden ist und daher kein freigegebener Autorisierungsdialog existiert, ist die Freigabe des Zugriffs leider etwas abschreckend, zumal die angeforderten Rechte sehr umfangreich sind, weil nicht nur gelesen, sondern auch geschrieben werden soll. Dennoch kann man sich im Code des Scripts jederzeit davon überzeugen, dass keine Daten den Browser in unerwartete Richtungen verlassen. Der Zugriff dient ausschließlich zur Pflege des Conversions Manager Tags und ggf. damit verbundenen Variablen und Trigger.

Im Verlauf der Autorisierung ist es deshalb leider notwendig, die erweiterten Optionen einzublenden, um dann per Klick auf einen weiteren Link zur ungeprüften, "unsicheren" Anwendung zu gelangen.



Diese Freigabe ist nur einmalig erforderlich, anschließend dann die angeforderte Funktion aus dem Menü ausgeführt werden.

Regeln bearbeiten

Nach Abschluss des Lesevorgangs wird das Tabellenblatt "*Conversion Rules*" eingeblendet. Sind keine Regeln vorhanden und soll das Tag komplett neu bestückt werden, kann der Schritt auch entfallen und man startet direkt auf diesem Arbeitsblatt mit der Definition.

Es enthält eine Tabelle, in der in einzelnen Spalten die Angaben aus der Tabelle "*Conversion data*" des Tags zu finden sind.

The screenshot shows the 'Conversions Manager Toolbox' interface. At the top is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Format', 'Data', 'Tools', 'Extensions', and 'Help'. Below the menu is a toolbar with various icons for search, undo, redo, print, copy, paste, zoom, and text formatting. The main area displays a table titled 'Pattern / Trigger Rule' with columns A through F. The table contains several rows of conversion rules, each with a unique pattern, a name, a value, a Google Ads label, a custom field, and a boolean 'fblead' status.

Pattern / Trigger Rule	Name	Value	Google Ads Label	Custom	fblead
re:\ueber-uns\	generate_lead	20	4adstest4		TRUE
ee:testEvent	generate_lead	0			TRUE
ee:purchase		{{ecommerce.value}}	1adstest1		
ee:add_to_cart		0	2adstest2		
cc:cu:mailto:	initiate_lead	5	3adstest3		TRUE
cc:cu:tel:	initiate_lead	5	3adstest3		TRUE
/produkt/atomkraftwerk-klein//produkt/atomkraf	small_fish	0			
re:.*impressum\.[?&]variant=[1 4].*	ab_test_group1	0			

Neue Zeilen können hier deutlich einfacher angelegt werden, denn alle Annehmlichkeiten der Google Tabellenkalkulation wie das Duplizieren von Zeilen etc. stehen dabei zur Verfügung.

Tabellenspalten statt JSON!

Besonders praktisch ist dabei, dass man im Feld "*Custom*" kein JSON eintragen muss, um es im Tag zu verwenden. Stattdessen können beliebig viele weitere Spalten rechts von "*Custom*" angelegt und mit Daten befüllt werden. Die Toolbox macht daraus selbst die passenden JSON Einträge und übergibt diese als Inhalt unter "*Custom*" beim späteren Export an das Tag.

Hinweis: Wenn sowohl in den weiteren Spalten, als auch im Feld "*Custom*" selbst etwas eingetragen ist, wird die Toolbox den Wert beim Schreiben des JSON Inhalts unter einem eindeutigen Namen (z. B. "*custom_value*") in das Ergebnis übernehmen, um keine Daten zu verlieren. Das ist aber natürlich nur ein Pflaster. Grundsätzlich ist es durchaus erlaubt, das *Custom* Feld für verschiedene Zeilen unterschiedlich zu nutzen, so dass eine Zeile dort einen Wert beinhaltet, aber die weiteren Spalten rechts davon leer lässt. Oder das *Custom* Feld bleibt leer und die anderen Spalten werden genutzt, um daraus ein JSON für "*Custom*" zu generieren. Beides gemischt in einer Regel zu nutzen, ist also nicht sinnvoll.

Zur Erinnerung: Bei der Generierung der Eingaben für "*Pattern / Trigger Rule*" hilft bei Bedarf das Tabellenblatt "*Pattern Helper*", auf dem die Präfixe anhand von Auswahllisten erstellt werden. Dabei ist allerdings das Mittel der Extraktionen (Präfixe *pc*, *pe*, *pr*) nicht abgedeckt, weil das die Bedienung dieses Hilfsmittels

unnötig kompliziert machen würde. Wer damit arbeiten will, wird aber sicher mit den Hinweisen aus dem Abschnitt zu Regeln und ohne eine Vorlage auskommen.

Um einen neuen Dienst hinzuzufügen, welcher weitere Daten aus dem Conversions Manager benötigt, muss nur eine neue Spalte angelegt und für alle betroffenen Conversions mit Daten versehen werden. Das ist im Spreadsheet deutlich angenehmer als über das UI des Tag Managers.

Tag aktualisieren

Ist die Bearbeitung abgeschlossen, wird im Menü die Funktion "*Write Updated Conversion Rules*" aufgerufen. Mehr ist nicht erforderlich, um das Tag im GTM Container mit den neuen Regeln auszustatten.

Bei der Erstellung des Inhalts für das *Custom* Feld werden dabei je Zeile nur die zusätzlichen Spalten berücksichtigt, die auch einen Wert enthalten. Das hält die Inhalte der jeweiligen Tabellenzeile im Tag so schlank wie möglich.

Nach Abschluss wird auf das Tabellenblatt "*Variables*" gewechselt. Hier sind alle Spaltennamen der Tabelle "*Conversion Data*" vom Tabellenblatt "*Rules*" Einträge in einer Tabelle zu finden, die den dataLayer Pfad des Variablenwertes, eine Vorgabe für einen Variablennamen und ggf. die ID der bereits bestehenden Variable aus dem ausweisen.

Von hier aus können noch fehlende Variablen für neue Felder ebenfalls automatisch angelegt werden.

Variablen und Trigger automatisiert erstellen

Dazu wird in der Spalte "*Create New Variable*" der Tabelle ein kleines "x" eingetragen, um diese zu markieren. Wurden gerade neue Felder erzeugt und das Tag im GTM damit aktualisiert, sind neue Variablen bereits eingetragen und markiert, so dass ein Aufruf von "*Create Selected Variables*" aus dem Menü ausreicht, um diese im Container per API erzeugen zu lassen.


	A	B	C	D	E	F
	Data Layer Variables					
	Key Name	dataLayer Path	Variable Name	Variable ID	Create New Variable	Create Trigger
	name	conversionInfo.name	conversionInfo.name	318		
	value	conversionInfo.value	conversionInfo.value	320		
	ads_label	conversionInfo.ads_label	conversionInfo.ads_label	319		
	fblead	conversionInfo.custom.fblead	conversionInfo.custom.fblead	314		
	solute	conversionInfo.custom.solute	conversionInfo.custom.solute		x	
	matomo_cat	conversionInfo.custom.matomo_cat	conversionInfo.custom.matomo_cat	316		
	matomo_action	conversionInfo.custom.matomo_action	conversionInfo.custom.matomo_action	317		
	lead_type	conversionInfo.custom.lead_type	conversionInfo.custom.lead_type	313		


Ist der Vorgang abgeschlossen, werden die IDs der neu angelegten Variablen in die dafür vorgesehene Spalte eingetragen und die “x” Markierungen entfernt, die zuvor für die Anlage der Variablen gesorgt haben.

Neue Variablen im Tag Manager haben den hier definierten Variablennamen, verweisen auf den Datenschicht-Pfad zum jeweiligen Feld und verwenden einen Standardwert, der dem aus den Tag-Einstellungen entspricht (also “*NONE*” als Vorgabe).

Variable konfigurieren

Variablentyp


Datenschichtvariable


Name der Datenschichtvariablen 

conversionInfo.custom.matomo_cat

Datenschichtversion

Version 2

✓ Standardwert festlegen

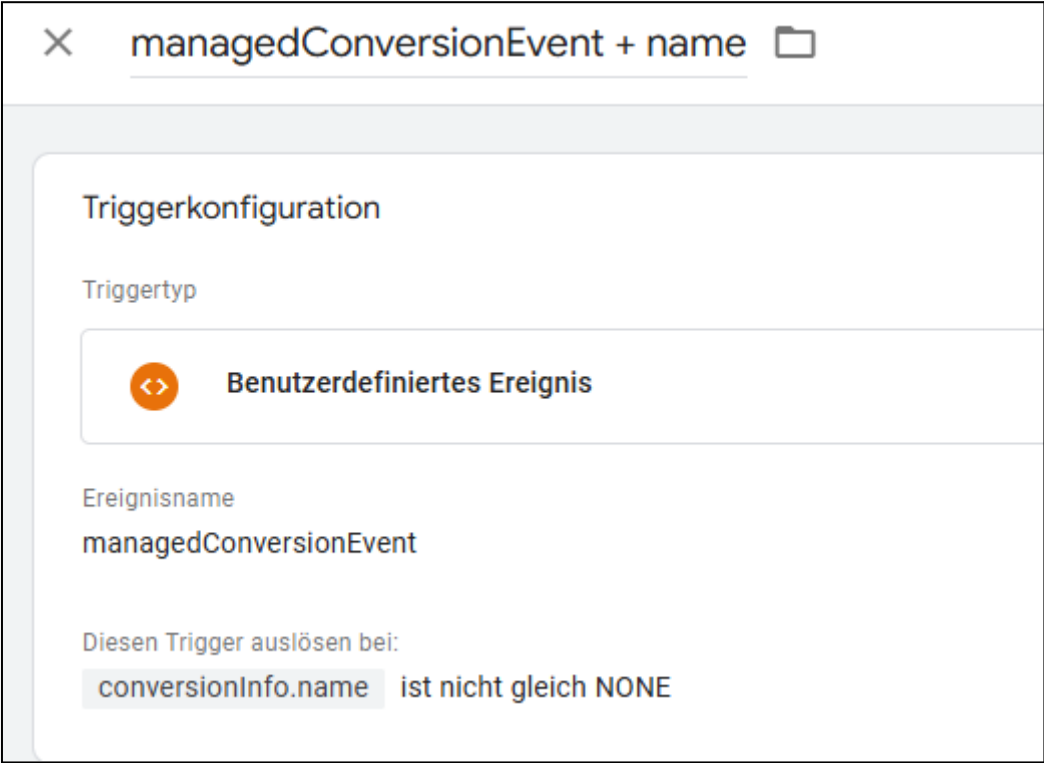
Standardwert 

NONE

Diese Vorgabe muss allerdings ggf. noch im GTM angepasst werden, wenn die entsprechenden Trigger z. B. nur dann auslösen sollen, wenn im Feld aus der Tabelle übernommen ein “*true*” als Wert zu finden ist und daher ggf. lieber “*false*” als Standardwert statt “*NONE*” verwendet werden sollte.


Als letztes Hilfsmittel kann wahlweise auch eine **automatisierte Anlage von Triggern** erfolgen. Dazu dient die letzte Spalte der Tabelle auf dem “Variables” Arbeitsblatt. Entsprechend der Anlage einer Variable kann über ein “x” auch in dieser Spalte die Erzeugung eines Triggers bestimmt werden. Im Menü dient “*Create Selected Triggers*” für die Anlage per API.

Die so neu angelegten Trigger haben einen Namen nach dem Muster *[conversionEventName] + [fieldName]*. Der *conversionEventName* kommt dabei aus den Einstellungen des Conversions Manager Tags, der *fieldName* entspricht dem Namen des Schlüssels der zugehörigen Variable im dataLayer. Legt man also z. B. für die Variable *conversionInfo.name* einen Trigger an und lautet der Name des dataLayer Events unverändert *managedConversionEvent*, entsteht der folgende Trigger:



The screenshot shows a web interface for configuring a trigger. At the top, there is a title bar with a close button (X), the text "managedConversionEvent + name", and a folder icon. Below this is a section titled "Triggerkonfiguration". Under the heading "Triggertyp", there is a selection box containing an orange icon with white code symbols (<>) and the text "Benutzerdefiniertes Ereignis". Below this, the "Ereignisname" is set to "managedConversionEvent". At the bottom, the text "Diesen Trigger auslösen bei:" is followed by a configuration rule: "conversionInfo.name" (highlighted in a light blue box) followed by "ist nicht gleich NONE".

Damit können alle Tags ausgelöst werden, die (ausschließlich) auf den Parameter “Name” aus den Conversiondaten angewiesen sind, um das Minimum an Daten zum Feuern zu haben. Im Beispiel also das GA4 Tag, welches nach gleichem Vorbild wie das zuvor abgebildete Beispiel-Tag für Google Ads den Namen als Variable verwendet. Dabei können auch andere Variablen genutzt werden, um optionale Felder des Tags mit Werten zu versorgen (wie den *value*).


Google Analytics: GA4-Ereignis
 Google Analytics

Mess-ID
 {{GA4 Measurement Id}}


Es kann nicht erkannt werden, ob sich das Google-Tag in Ihrem C befindet
 Eventuell haben Sie eine Variable verwendet oder einen ungültigen eingegeben. [Weitere Informationen](#)

Ereignisname ⓘ
 {{conversionInfo.name}}


Ereignisparameter
 Event Settings Variable
 {{Shared Event Settings}} ⓘ

Ereignisparameter		Wert
element_type		{{conversionInfo.custom.lead_type}}
value	✓	{{conversionInfo.value}}
currency	✓	EUR
something		{{Demo Custom String - Erster Eintrag}}


Trigger
 Auslösende Trigger


managedConversionEvent + name
 Benutzerdefiniertes Ereignis

Ausnahmen


Blocking | No Analytics
 Benutzerdefiniertes Ereignis

ODER


Blocking | No Preview
 Benutzerdefiniertes Ereignis

Beispielrechnung zu Einsparungen

Um zu bestimmen, ob sich der ganze Aufwand lohnt oder nicht, kann man eine einfache Rechnung anstellen. Davon ausgehend, dass es zehn verschiedene Conversions oder Events gibt, die alle an drei unterschiedliche Empfänger gesendet werden müssen und alle dafür erforderlichen Tags sind derzeit einzeln im Tag Manager Container implementiert, umfasst dieses "reguläre" Conversion- und Event-Tracking Setup insgesamt **40 Elemente**:

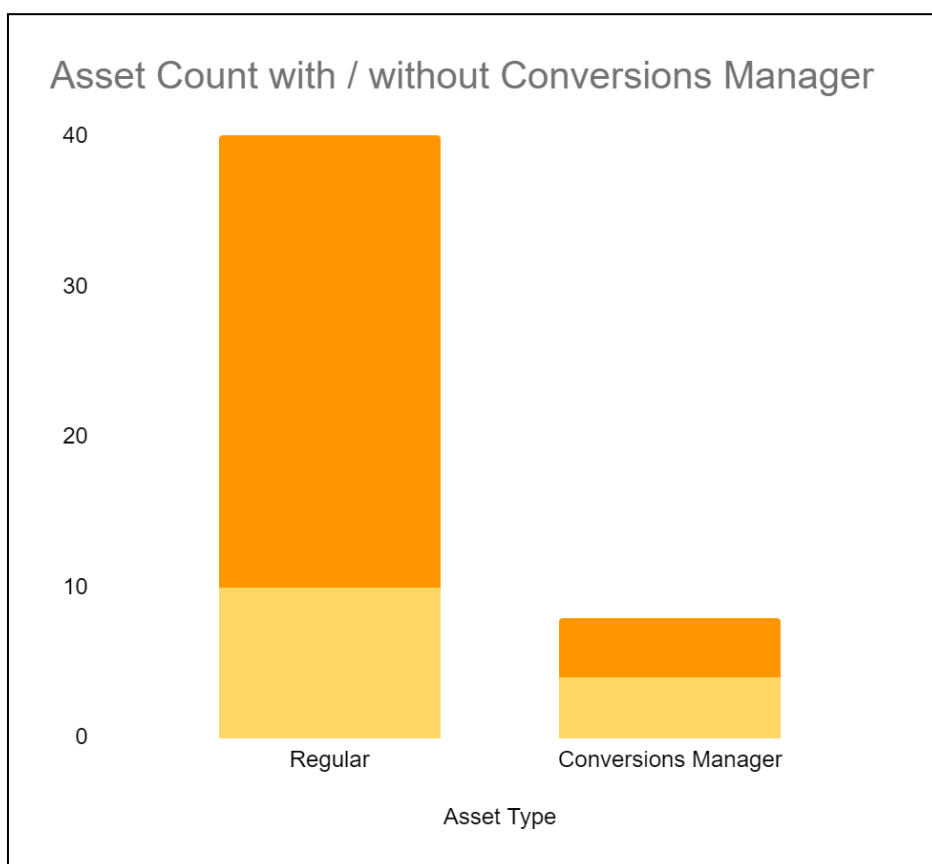
- 10 feuernde Trigger

- 30 Tags (10 pro Dienst / Empfänger)

Stellt man dies nun auf den Conversions Manager um, verbleiben nur noch **8 Elemente**, wenn man mit blockierenden Triggern arbeitet (was die meiste Flexibilität liefert):

- 1 feuernder Trigger
- 3 blockierende Trigger
- 4 Tags (3 Dienste + Conversions Manager Tag)

Wer sich von Bildern statt Zahlen besser überzeugen lässt:



Die zusätzlich angelegten Variablen für den Zugriff auf die Eigenschaften aus dem Conversions Manager sind dabei ehrlicherweise nicht mitgezählt worden, fallen in der Bilanz eines Containers aber auch deutlich weniger auf als Tags. Mit kombinierten Triggern kann das Ergebnis sogar noch **ein Element weniger** beinhalten.

Wer nach dieser Rechnung entscheidet, den Conversions Manager einzusetzen, um den Container mit einer zentralen Steuerung zu versehen und / oder bei

zukünftigen Änderungen auf dieses Mittel zurückgreifen zu können, hat mit den Hinweisen aus den vorherigen Abschnitten alle Bausteine in der Hand.

Danke!

Hast Du Anmerkungen? Dann freue ich mich über eine Nachricht an mail@markus-baersch.de.

Danke fürs Lesen und viel Spaß!

Markus Baersch

November 2024