



## Splunk Knowledge Manager 説明書

バージョン : 4.0.3

作成日 : 8/26/2009 09:59 am

Copyright Splunk, Inc. All Rights Reserved

# 目次

<b>はじめに</b> .....	<b>1</b>
このマニュアルについて.....	1
このマニュアルの対象者.....	1
Splunk ナレッジとは.....	1
<b>初めて Splunk を使う場合</b> .....	<b>3</b>
インデックスのしくみ.....	3
インデックスタイムと検索時間.....	4
<b>イベントの理解</b> .....	<b>6</b>
イベントについて.....	6
イベントタイムスタンプの概要.....	6
イベントセグメンテーションの概要.....	7
複数行イベントおよびイベント改行コードの概要.....	8
デフォルトフィールド抽出の概要.....	9
<b>フィールドの取り扱い</b> .....	<b>10</b>
フィールドについて.....	10
検索時間でフィールドの追加.....	11
検索時間フィールド抽出の管理.....	15
インデックスタイムフィールド抽出のカスタマイズ.....	18
外部データソースのフィールド検索.....	22
ソース入力時にファイルヘッダーからフィールドを抽出.....	27
複数の値を持つフィールドの設定.....	32
<b>ホストの取り扱い</b> .....	<b>33</b>
ホストについて.....	33
デフォルトの Splunk サーバーホストの設定.....	34
入力に対するホスト割り当ての設定.....	35
イベントデータを基にしたデフォルトホスト割り当ての上書き.....	38
<b>ソースタイプの取り扱い</b> .....	<b>41</b>
ソースタイプについて.....	41
ソースタイプの名前変更.....	43
ルールベースのソースタイプ認識の設定.....	43
Splunk のソースタイプ自動分類子の教育.....	45
教育済みソースタイプ.....	45
ソースタイプ自動割当の回避.....	50

<a href="#">props.conf でソースタイプ設定を指定</a>	<a href="#">52</a>
<b><a href="#">イベントタイプの管理</a></b>	<b><a href="#">54</a></b>
<a href="#">イベントタイプについて</a>	<a href="#">54</a>
<a href="#">Splunk Web によるイベントタイプの定義</a>	<a href="#">56</a>
<a href="#">eventtypes.conf に直接イベントタイプを設定</a>	<a href="#">56</a>
<a href="#">イベントタイプテンプレートの設定</a>	<a href="#">58</a>
<b><a href="#">タグとエイリアスの定義</a></b>	<b><a href="#">59</a></b>
<a href="#">タグとエイリアスについて</a>	<a href="#">59</a>
<a href="#">フィールドのエイリアス作成</a>	<a href="#">59</a>
<a href="#">ホストフィールドのタグ付け</a>	<a href="#">60</a>
<a href="#">イベントタイプのタグ</a>	<a href="#">61</a>
<b><a href="#">イベントをトランザクションにグループ化</a></b>	<b><a href="#">63</a></b>
<a href="#">トランザクションについて</a>	<a href="#">63</a>
<a href="#">トランザクションの定義</a>	<a href="#">65</a>
<b><a href="#">保存済み検索と検索ジョブの管理</a></b>	<b><a href="#">68</a></b>
<a href="#">保存済み検索の管理</a>	<a href="#">68</a>
<a href="#">マクロ検索の設計</a>	<a href="#">68</a>
<a href="#">フォーム検索の設計</a>	<a href="#">69</a>
<a href="#">保存済み検索とレポートのナビゲーションの定義</a>	<a href="#">69</a>
<b><a href="#">サマリーインデックスの設定</a></b>	<b><a href="#">71</a></b>
<a href="#">サマリーインデックスの設定</a>	<a href="#">71</a>

# はじめに

## このマニュアルについて

### このマニュアルについて

このナレッジマネージャマニュアルは、利用者のニーズを満たすよう Splunk でデータを処理するための適切な使い方および拡張の仕方について説明しています。

本書は、Splunk の展開を最適化、維持、拡張する際にご利用ください。

本書には、Splunk の展開をスムーズに行う助けになる、手順に沿った使用方法のみでなく、概念的に重要な情報がたくさん記載されています。以下について説明します。

- Splunk のインデックスのしくみ
- イベント、イベントタイプ、フィールド、ソースタイプ、タグ、トランザクションなどの Splunk「知識オブジェクト」を管理維持する方法
- フィールドの適切な処理の仕方
- 関連するイベントを的確にトランザクションにグループ化する方法

新しいこのマニュアルおよび本マニュアルの対象者について知るためにこの先をお読みください。

## このマニュアルの対象者

### このマニュアルの対象者

本書には、Splunk ナレッジマネージャに関する情報および使用手順が記述されています。あなたが、ご自分および会社内の別のユーザーのために Splunk のデータおよび知識を維持し拡張する必要があるパワーユーザーの場合、あなたはナレッジマネージャです。

他の人が使うために保存済み検索の作成、トランザクションの定義、カスタムフィールドの追加、タグの管理および生成、またはデータの処理を向上させるためにインデックス設定の変更を行う場合は、本書を参照してください。

## Splunk ナレッジとは

### Splunk ナレッジとは

Splunk は、IT データの詳細と大きなパターンの両方を見る助けとなるパワフルツールです。Splunk を使うとき、ログファイルの個別エントリを単に見るのみでなく、その情報を活用して環境について正しく知ることができます。

そのためには、Splunk ナレッジを作成し使用します。デフォルトでは、Splunk が **フィールド**、**ソースタイプ**、**イベントタイプ**などの知識をデータに追加します。それらを編集して追加できます。

利用者の定義をする Splunk ナレッジとは、**タグ**、**保存済み検索**、および**トランザクション**を含みます。

本章では、様々な Splunk ナレッジについての概要を記述しています。以下の章では、これらのナレッジを管理および処理するための具体的な方法を紹介します。

- イベントについて
- フィールドについて
- ソースタイプについて
- イベントタイプについて
- タグについて
- トランザクションについて

# 初めて Splunk を使う場合

## インデックスのしくみ

### インデックスのしくみ

インデックスは、Splunk が、ユーザーが送信したデータを処理して、検索および分析する手段です。Splunk は、あらゆるタイプの時間列データ(タイムスタンプの付いたデータ)にインデックスを付けることができます。Splunk がデータにインデックスを付けると、タイムスタンプを基にイベントに分類されます。

Splunk は、インデックス用のイベントデータ(イベントに対する各種アクションを実行)を処理します。

- イベントにタイムスタンプがない場合は、Splunk が作成しようとします。Splunk は、タイムゾーンオフセットを適用して欧州日付形式を認識するように設定できます。
- イベントはすべて、検索可能なセグメントに分解されます。インデックスおよび検索速度、検索機能、ディスク圧縮効率に影響するセグメントのレベルを決めることができます。
- イベントの多くは短く、長くても 1 行または 2 行ですが、それより長いイベントもあります。Splunk は、改行ルールを採用して検索結果を表示する際のイベントの改行規則を決めます。
- Splunk は、イベントのホスト、ソース、ソースタイプなどを含む各イベントのデフォルトフィールドを抽出して受信イベントデータを処理します。
- Splunk は、インデックス処理中に機密のイベントデータ（クレジットカードまたは XX 番号など）を匿名化するように設定できます。カスタムメタデータを受信イベントに適用するように設定することもできます。
- イベントおよびインデックス処理中のイベントの動作については、本書の「イベントについて」を参照してください。
- インデックスは I/O 集中型プロセスです。

### インデックスとは？

Splunk は、インデックスで処理するすべてのデータを保管します。インデックスは、データベース (\$SPLUNK\_HOME/var/lib/splunk) に保管されます。データベースは、db\_<starttime>\_<endtime>\_<seq\_num> という名前のディレクトリです。インデックスは、データベースディレクトリを集めたものです。

Splunk には、予め設定された以下のインデックスが付いています。

- **main:** これはデフォルトの Splunk インデックスです。指定しない限り、処理したデータはすべてここに保存されます。
- **splunklogger:** Splunk はこのインデックスで内部ログの追跡を保存します。
- **\_internal:** Splunk の処理メトリクスを保存します。
- **sampledata:** トレーニング用の少量のサンプルデータがここに保存されます。
- **\_thefishbucket:** 情報を処理する内部ファイルを格納します。
- **\_audit:** ファイルシステム変更監視、監査、全ユーザーの検索履歴などに関するイベントを格納します。

Splunk 管理者は、新規インデックスの作成、インデックスプロパティの編集、不要なインデックスの削除、既存のインデックスの配置換えなどが行えます。 Splunk の管理者は、Splunk 管理、CLI、indexes.conf などの設定ファイルを使ってインデックスを管理します。詳しくは、管理者マニュアル の「インデックスの管理」を参照してください。

## インデックスタイムと検索時間

### インデックスタイムと検索時間

Splunk の説明書には、インデックスタイムと検索時間という用語が頻繁に使用されています。 これらの用語は、Splunk でインデックスを付けるときに処理されるイベントデータの種類と検索が実行されるまで実際には存在しないイベントデータの種類を区別するために使用されています。

ユーザー用に作成し管理する知識オブジェクトに関する意思決定に影響するため、知識マネージャがこの区別を理解することが重要です。

例えば、データにまだインデックスが付けられていない状態で、カスタマイズされたソースタイプとホストを大量に持つ予定がある場合は、インデックス付けを開始する前にこれらのソースタイプとホストを知っておきたい場合があります。この作業は、カスタムソースの種類とホストを決め ( ルールベースのソースタイプの割り当て、ソースタイプの上書き、入力ベースのホスト割り当て、ホストの上書きなどを使用 ) て、インデックス処理中に処理できるようにします。インデックス付けが完了した後は、ホストまたはソースタイプの割り付けを変更できませんが、別の値でタグ付けして問題を管理できます。

### 作業中のトピック

#### インデックスタイム

インデックスタイムの処理は、イベントデータに実際にインデックスが付けられる前に行われます。

インデックスタイム中 (または前) に以下のプロセスが実行されます。

- ヘッダーベースのフィールド自動抽出
- 特定の入力に対する静的または動的なホストの割り当て
- デフォルトホスト割り当ての上書き
- ソースタイプのカスタマイズ
- イベントのタイムスタンプ付け
- イベントの改行処理
- イベントのセグメント分解(検索字でも発生)
- デフォルトフィールドの抽出(host、source、sourcetype、timestamp など)

## 検索時間

検索時間の処理は、検索でイベントが正しく収集されたなど、検索を実行した後に行われます。検索時間には、以下の処理が行われます。

- セグメント分解 (インデックスタイムでも発生)
- イベントタイプの一致
- 検索時間フィールドの抽出 (multivalued フィールド構文分解など、自動およびカスタムフィールド抽出を含む)
- フィールドエイリアシング
- 外部データソースのフィールドを検索
- ソースタイプの名前変更
- タグ付け

# イベントの理解

## イベントについて

イベントについて

イベントとは、ログファイルが付いたアクティビティの記録で、主に Splunk により付けられたものを言います。ログファイルを生成したシステムに関する情報を提供します。特に、インデックスプロセスの出力を「イベントデータ」と呼びます。

例えば、

```
172.26.34.223 - - [01/Jul/2005:12:05:27 -0700] "GET /trade/app?action=logout HTTP/1.1" 200 2953
```

Splunk でイベントにインデックスを付けると、

- イベントのタイムスタンプを特定する(および、存在しない場合は、イベントにタイムスタンプを適用する)
- イベント分解の実行
- 複数ラインのイベントを識別し、必要に応じて改行を実行
- 便利な標準フィールド(host、source、sourcetype など)の抽出

ここでは、これらの動作とそれに関する詳細の見つけ方について簡単な概要を説明します。

Splunk のインデックス処理の概要については、管理者マニュアルの「インデキシングとイベント処理」章を参照してください。

## イベントタイムスタンプの概要

イベントタイムスタンプの概要

「イベントについて」で紹介したサンプルイベントをご覧ください。

```
172.26.34.223 - - [01/Jul/2005:12:05:27 -0700] "GET /trade/app?action=logout HTTP/1.1" 200 2953
```

これには次のイベントの時間情報が含まれています。 [01/Jul/2005:12:05:27 -0700]。これが**タイムスタンプ**と呼ばれています。Splunk は、タイムスタンプを使ってイベントを時間に関連付け、Splunk Web でヒストグラムを作成し、検索用の時間範囲を設定します。ほとんどのイベントには、タイムスタンプが含まれています。タイムスタンプ情報が含まれていない場合、Splunk がインデックスを付ける際にタイムスタンプ値を割り当てようとします。

イベントのほとんどは、タイムスタンプフォーマットの処理を加える必要はありませんが、Splunk 管理者が設定を行う必要がある場合があります。例えば、Splunk の管理者がタイムスタンプの識別およびフォーマットを再設定する必要がある場合など、ソースおよび分散展開の場合が挙げられます。この他にも、以下の場合に管理者がタイムスタンプを処理することがあります。

- 高度なインデックス処理を行うためのタイムスタンプ抽出の調整
- 複数タイムスタンプを持つイベントのタイムスタンプ抽出の設定
- タイムスタンプオフセットのアプリケーション(異なるタイムゾーンにおけるイベントの関連付け)
- ローカライズされたタイムスタンプ形式(ヨーロッパ用など)を Splunk で識別できるようにする

このトピックについては、管理者マニュアルの「タイムスタンプ」章を参照してください。

## イベントセグメンテーションの概要

### イベントセグメンテーションの概要

**セグメンテーション**は、インデックスタイムおよび検索タイムに、イベントを検索可能なセグメントに分割するために Splunk が使用します。セグメントは**メジャー**または**マイナー**で区別されます。簡単には、メジャーセグメントをマイナーセグメントで分割できます。例えば、IP アドレス `172.26.34.223` は、全体がメジャーセグメントです。ただし、このメジャーセグメントは、`172` のようなマイナーセグメントおよび `172.26.34` のようなグループとしてのマイナーセグメントに分割できます。

Splunk を使うと、Splunk 管理者がイベントセグメンテーションの仕方を定義できます。これは、**インデックスタイムセグメンテーション**がインデックスおよび検索速度、ディスク圧縮、および先行入力機能の使用に影響を及ぼすため重要です。検索タイムセグメンテーションも、Splunk Web の表示結果から項目を選択して検索する速度と検索を作成する機能に影響します。

インデックスタイムセグメンテーションは、`segmenters.conf` を使って設定します。検索タイムセグメンテーションは、Splunk Web 検索アプリケーションのインタフェースをから開く**オプションポップアップ**で設定します。

「インデックスタイム」および「検索タイム」の詳細は、本書の「インデックスタイムと検索タイム」を参照してください。

### イベントセグメンテーションのレベル

インデックスタイムと検索タイムで管理者が使えるセグメンテーションには以下の3つのレベルがあります。

- **内部セグメンテーション**は、イベントを可能な限り小さなセグメントに分解します。例えば、`172.26.34.223` などの IP アドレスは、内部セグメンテーションを使って `172`、`26`、`34`、`223` などのセグメントに分解されます。インデックスタイムで内部セグメンテーションを設定すると、検索速度に関しては非常に効率的なインデックスが行えますが、インデックスの速度に影響を与え、先行入力機能を制限します。(マイナーセグメントレベルでのみ先行入力機能が使用可能です。)
- **外部セグメンテーション**は内部セグメンテーションの反対です。外部セグメンテーションでは、メジャーセグメントのみがインデックスされます。そのため、IP アドレスはコンポーネントに分割されません。インデックスタイムで外部セグメンテーションを設定した場合は、ワイルドカードを使わなければ IP アドレスを個別に検索できません。外部セグメンテーションで作成されたインデックスは、フルセグメンテーションで作られたものより多少効率が良くなりますが、内部セグメンテーションで作成されたインデックスより効率が良くありません。
- **フルセグメンテーション**は、内部および外部セグメンテーションを組み合わせた特性を持ち合わせます。フルセグメンテーションを使うと、IP アドレスは、メジャーセグメントと各種マイナーセグメント(`172.26` と `172.26.34` の組み合わせを含む)の両方でインデックスされます。これは、最も効率の悪いのインデックスオプションですが、最も多様性のある検索用語を提供します。

**注記：** デフォルトでは、インデックスタイムセグメンテーションは、内部および外部セグメンテーションの組み合わせで設定されますが、検索タイムセグメンテーションはフルセグメンテーションで設定されます。

セグメンテーションのレベル変更については、管理者マニュアルの「セグメンテーションを設定してディスク使用を管理」を参照してください。

特定のホスト、ソース、ソースタイプに対してセグメントルールを定義する

Splunk 管理者は、特定のホスト、ソースまたはソースタイプを持つイベントに特別に適用するインデックスタイムおよび検索タイムセグメンテーションルールを定義できます。定期的に特定のソースタイプに対して検索を実行する場合、この機能を使用して、検索性能を向上させることができます。同様に、大量の `syslog` イベントを頻繁にインデックスする場合は、この機能を使ってイベントが使う全体的なディスクスペースを減らす役に立ちます。

これら特定のセグメンテーションルールを設定する方法に関する詳細は、管理者マニュアルの「ホスト、ソース、またはソースタイプのカスタムセグメンテーションの設定」を参照してください。

## 複数行イベントおよびイベント改行コードの概要

複数行イベントおよびイベント改行コードの概要

イベントには 1 行以上で構成されるものがあります。Splunk は、ほとんどイベントをデフォルトで正しく処理しますが、デフォルトで適切に認識できない複数行のイベントがある場合があります。

Splunk の改行コード処理のデフォルト設定を変更する方法については、管理者マニュアルの「複数行イベントのインデックス」を参照してください。

複数行イベントの改行コード処理とセグメンテーションの制約

大量のイベントに改行コードおよびセグメンテーションを行うと、Splunk に制約が適用されます。

- **10,000 バイト以上の行：** Splunk は、インデックスする際に 10,000 バイトを超える行を 10,000 バイト毎に改行して複数行にします。複数行の各行の最後に `meta::truncated` フィールドを付加します。ただし、複数行でも 1 つのイベントグループとして処理します。
- **100,000 バイト以上のイベントに対するセグメンテーション：** Splunk では、イベントの最初の 100,000 バイトのみを検索結果に表示します。ただし、長い行の最初の 100,000 バイト以降のセグメントも検索可能です。
- **1,000 セグメント以上のイベントに対するセグメンテーション：** Splunk は、1 つのイベントの個別の最初の 1,000 セグメントを空白文字で区切り、マウスを上に移動させたときにハイライトしてセグメントとして表示します。このとき、イベントの残りの部分は、インタラクティブな形式を持たないローデータで表示します。

# デフォルトフィールド抽出の概要

## デフォルトフィールド抽出の概要

Splunk がイベントデータをインデックスするとき、ほとんどのイベントで共通する一連のフィールド、つまりの検索およびレポートで共通に使用するフィールドをデフォルトで抽出します。デフォルトのフィールドには以下が含まれます。

- `host`: 発信元ホスト名またはイベントを生成したネットワークデバイスの IP アドレスを特定します。生成した特定のホストを持つイベントの検索の絞り込みに使用します。
- `source`: イベントがインデックスされたファイル名またはパス名を特定します。検索するイベントを絞り込む、またはデータ処理コマンドの引数に使用します。
- `sourcetype`: `access_log` または `syslog` などイベントが表すアプリケーション、ネットワークまたはデバイスデータのタイプを特定します。Splunk 管理者は、予めソースの種類を定義することができます。または、Splunk がインデックスを付加する際に自動的に生成することもできます。 `sourcetype` を使って検索するイベントを絞り込む、または `sourcetype` をデータ処理コマンドの引数に使用します。

インデックス処理で Splunk が特定するデフォルトフィールドの一覧および、検索で使用方法については、ユーザーマニュアルの「デフォルトと内部フィールドの使用」を参照してください。

## 追加フィールドの抽出

Splunk では、インデックスタイムで特定されたデフォルトフィールドおよび検索時間に自動的に抽出されたフィールドが十分でない場合、追加のフィールドを抽出できます。Splunk ナレッジマネージャとして、これらのカスタムフィールドを作成して、組織のニーズに特化した、重要なイベント情報を追跡できます。詳しくは、本書の「イベントの理解」の項を参照してください。ここでは、以下について学びます。

- Splunk Web または設定ファイルを使用した検索時間のカスタムフィールドの抽出
- デフォルトフィールド抽出のインデックスタイムのカスタマイズ(推奨はしませんが、必要になる場合があります)
- 外部データソースのフィールド検索の作成
- ソース分類処理中にヘッダー付きファイル(CSV および MS Exchange ファイルなど)からカスタムフィールドを抽出
- フィールドのエイリアス作成
- マルチバリューフィールドの設定

# フィールドの取り扱い

## フィールドについて

フィールドについて

フィールドは、イベントデータにある検索可能な名前と値のペアです。フィールドは、フィールドで処理されるすべてのイベントを作るインデックスされたセグメントと区別され、名前を持ち、その名前で検索可能です。

例えば、以下の検索を見てみましょう。

```
host=foo
```

この検索では、`foo` の値を持つ `host` フィールドのイベントを検索する方法を `host=foo` で示しています。この検索を実行すると、Splunk は、異なる `host` フィールド値を持つイベントは検索しません。また、`foo` を値として共有するその他のフィールドを含むイベントも検索しません。つまり、この検索では、検索バーに単に `foo` を入力した場合より焦点を絞った検索結果が出ます。

Splunk がイベントデータを処理する際、まずインデックスタイムで、次に検索時間で自動的にフィールドを抽出および定義します。

- **インデックスタイム**では、`host`、`source`、`sourcetype` などを含む各イベントの小規模なデフォルトフィールドを抽出します。デフォルトフィールドはすべてのイベントに共通です。
- **検索時間**では、イベントデータから幅広い範囲のフィールドを特定して抽出します。例えば、`user_id` および `client_ip` フィールドの例としてそれぞれ `user id=jdoe` または `client ip=192.168.1.1` など、明確なフィールド名/値ペアを検索します。

カスタムフィールドの追加と維持

Splunk の IT 検索を完全に活用するためには、カスタムフィールドの追加および維持の方法を知る必要があります。カスタムフィールドを使うと、ニーズに特化した重要な情報を読み出して追跡できます。ナレッジマネージャは、組織の他の Splunk ユーザーが使用する特殊なカスタムフィールドを定義できます。ナレッジマネージャマニュアルのこのセクションでは、フィールドを作成し、維持するさまざまな方法について、およびこの機能の使い方を、例を挙げて説明しています。

ここでは、以下について学びます。

- 検索時間で新規フィールドの追加
- インデックスタイムフィールド抽出のカスタマイズ
- 外部データソースのフィールド検索
- ファイルヘッダーを基準にしたインデックスタイム抽出の設定
- マルチバリューフィールド構文解析の設定
- フィールドのエイリアス作成

# 検索時間でフィールドの追加

## 検索時間でフィールドの追加

Splunk を使用中、Splunk がインデックスタイムおよび検索時間で自動的に検索する一連のフィールドに追加する形となる新しいフィールドの作成が必要となる状況に直面する場合があります。ナレッジマネージャは、チームメンバーのためにフィールド抽出を管理する立場にあります。例えば、Splunk ナレッジマネージャは、イベントデータ標準化戦略の一部としてフィールド抽出を活用し、既存のフィールドを再定義したり、新しいフィールドを作成したりして、冗長性を減らし、チーム内の他の Splunk ユーザーがフィールドを使用する上で全体的な利便性を上げる取り組みをします。

Splunk が自動的に特定したフィールドの他に新しくフィールドを作成する必要がある場合、その実施にはいくつかの方法があります。フィールド抽出に使用できる Splunk Web の機能はたくさんありますが、設定ファイルの編集という方法により Splunk のバックエンドで抽出したフィールドを追加および管理することができます。

ここでは、Splunk Web のフィールド抽出の概要を簡単に説明し、設定ファイルによるフィールド抽出の管理について詳細を紹介します。

## Splunk Web を使った検索時間のフィールド追加

Splunk Web の機能を使った検索時間のフィールド追加に関する詳細は、ユーザーマニュアルの「新しいフィールドの抽出と追加」を参照してください。ここでは、概要を説明します。

## インタラクティブなフィールド抽出の使用

Splunk Web の対話式フィールド抽出機能 (IFX) を使ってカスタムフィールドを同時に作成できます。IFX を使うと、あらゆる検索を 1 つ以上のフィールドで行うことができます。ローカルインデックスマシンで IFX が使えます。IFX の使用については、ユーザーガイドの「Splunk Web で対話式にフィールドを抽出」を参照してください。

IFX にアクセスするには、検索を実行して、フィールド検索結果のタイムスタンプの下に表示されるドロップダウンから「フィールドの抽出」を選択します。IFX では、1 度に 1 つのフィールドのみを抽出することができます(正規表現構文を編集して、後で複数のフィールドを抽出できます)。

## 検索コマンドの使用

Splunk には、さまざまな方法でフィールドを抽出するための各種検索コマンドがあります。ここでは、そのコマンドを一覧しますが、その詳細および使用例については、検索リファレンスまたはユーザーマニュアルの「新しいフィールドの抽出と追加」を参照してください。

- ・ `rex` 検索コマンドは、検索文字列に含めたグループを指定する Perl の正規表現を使ってフィールドの抽出を行います。
- ・ `extract` (または「key/value」用 `kv`) 検索コマンドは、検索結果から強制的にフィールドと値を抽出します。引数を指定しないで `extract` を使うと、Splunk は `props.conf` に追加されたフィールド抽出文字列(スタンザ)を使ってフィールドを抽出します。`extract` を使って手作業で `conf` ファイルに追加したフィールド抽出をテストできます。

- multikv を使って、複数ライン、表形式のイベントからフィールドおよび値を抽出します。このコマンドは、各表の行に対して新しくイベントを作成し、表のタイトルでフィールド名を付けます。
- xmlkv は、ウェブページのトランザクションなど、xml 形式のイベントデータからフィールドおよび値を強制抽出します。
- kvform は、予め定義され、\$SPLUNK\_HOME/etc/system/form/ または、カスタムアプリケーションのディレクトリ \$SPLUNK\_HOME/etc/apps/ に保存されているフォームテンプレートを基に、フィールド/値ペアでイベントを抽出します。例えば、form=sales\_order の場合、Splunk は、sales\_order.form を検索して、このフォームに対して処理されたすべてのイベントの値を抽出しようとしています。

#### Splunk でフィールド名を作成する場合

Splunk で指定できるフィールド名は、以下のアルファベット文字またはアンダーラインのみです。

- フィールド名に指定できる文字 : a-z, A-Z, 0-9, \_
- フィールド名の最初の文字に 0-9 または \_ は指定できません。アンダーライン(\_)から始まる名前は、Splunk の内部変数に使用されています。
- 国際文字は使用できません。

Splunk では、インデックスタイムまたは検索時間による抽出に関わらず、デフォルトまたはカスタム設定で以下の規則を適用しています。

1. a-z、A-Z、0-9 の範囲外のすべての文字は、アンダーライン(\_)に置き換えられます。
2. 文頭のアンダーラインはすべて削除されます。文頭に 0-9 文字を使うとエラーになります。

#### 設定ファイル編集による検索時間のフィールド追加

ナレッジマネージャの多くは、設定ファイルを通してカスタムフィールドを管理するのがより簡単だと感じています。設定ファイルでは、チームメンバーが使用するカスタムフィールドの追加、維持、およびライブラリの閲覧ができます。

\$SPLUNK\_HOME/etc/system/local/、または \$SPLUNK\_HOME/etc/apps/ の独自のカスタムアプリケーションディレクトリで編集する props.conf に検索時間フィールドの抽出を追加します。(カスタマイズしたデータを別のサーバーに簡単に転送したい場合は、後者を使用してください。)

**注記：** \$SPLUNK\_HOME/etc/system/default/ のファイルは編集しないでください。

設定ファイルの全般的な内容については、管理者マニュアルの「設定ファイルについて」を参照してください。

ご存知のとおり、Splunk は、正規表現(regexes)を使ってイベントデータからフィールドを抽出します。IFX を使う場合、Splunk は正規表現を生成しますが、これでは一度に 1 つのフィールドしか抽出しません。逆に、設定ファイルを通じて手作業でフィールド抽出を設定すると、正規表現を自分で指定しなければなりません、必要に応じて複数のフィールドを抽出する正規表現を設定できます。

**重要：** 正規表現でグループを読み出す場合は、英数字文字またはアンダーラインを含むフィールド名を特定しなければなりません。

- フィールド名に指定できる文字 : a-z, A-Z, 0-9, \_
- フィールド名の最初の文字に 0-9 または \_ は指定できません。(アンダーライン ( \_ ) から始まる名前は、Splunk の内部変数に使用されています。)
- 国際文字は使用できません。

#### カスタム検索時間によるフィールド抽出設定の基本手順

1. イベントのフィールドを特定するパターンを指定します。
2. イベントからフィールドを抽出する正規表現を記述します。 rex 検索コマンドを使った検索を実行して正規表現をテストできます。
3. props.conf に正規表現を追加して、ソース、ソースタイプ、またはフィールドを検出したいイベントを含むホストにリンクします。
4. フィールド値が単語の一部の場合は、fields.conf にエントリーを追加する必要があります。下の例「サブトークンからフィールドを作成」を参照してください。

\$SPLUNK\_HOME/etc/system/local/、または \$SPLUNK\_HOME/etc/apps/ の独自のカスタムアプリケーションディレクトリにある transforms.conf および props.conf ファイルを編集します。

**注記 :** \$SPLUNK\_HOME/etc/system/default/ のファイルは編集しないでください。

5. Splunk を再起動して変更を有効にします。

#### props.conf に正規表現スタanzasを追加

フィールド抽出スタanzasを props.conf に追加する場合は、この形式を使います。

```
[<spec>]
EXTRACT-<class> = <your_regex>
```

- <spec> は以下が使えます。
  - ◆ <sourcetype>、イベントのソースタイプ。
  - ◆ host::<host>、<host> はイベントをホスト。
  - ◆ source::<source>、<source> はイベントのソース。
- <class> は抽出クラス。 クラスの優先順位規則 :
  - ◆ 各クラスに対して、Splunk は、最優先設定ブロックからの設定を受けます。
  - ◆ ある source および sourcetype に対して特定のクラスが指定されている場合は、source に対するクラスが優先されます。
  - ◆ 同様に、特定のクラスが <spec>用の ../local/ for a に指定されている場合は、../default/ のクラスを上書きします。
- <your\_regex> = は、カスタムフィールド値を識別する正規表現を作ります。各グループは異なる抽出フィールドを示すため、正規表現には、グループを読み出す名前が必要です。

**注記 :** インデックスタイムに Splunk が抽出する一連のデフォルトフィールドの設定手順と違い、検索時間フィールド抽出ではインデックスに書き込まれないため、 transforms.conf には、DEST\_KEY は必要ありません。検索時間で抽出されたフィー

ルドは、インデックスのキーとして存続しません。

**注記：** 検索時間フィールド抽出の場合、`props.conf` は、`TRANSFORMS-<value>` ではなく `EXTRACT-<class>` をインデックスタイムのフィールド抽出の設定に使用します。

#### 検索タイムフィールド抽出例

ここでは、設定ファイルを使って設定する、手動のフィールド抽出の例を紹介します。

#### 新しいエラーコードフィールドの追加

この例では、新しい「エラーコード」フィールドを作成する方法を紹介します。このフィールドは、`device_id=` に続く括弧内の単語とコロンの組み合わせで終結するテキスト文字列により特定できます。このとき、`testlog` ソースタイプに関連するイベントからフィールドが抽出されます。

`props.conf` に以下を追加します。

```
[testlog] EXTRACT-<errors> = device_id=\[w+\](?<err_code>[^\:]+)
```

#### 1つの正規表現で複数フィールドを抽出

ここでは、5つの異なるフィールドを引き出すフィールド抽出の例を紹介します。その後、これらのフィールドをいくつかのイベントタイプと協調させてポートがフラッピングしているイベントを探し、レポートするのに役に立ちます。

以下は、フィールドが抽出されたイベントデータのサンプルです。

```
##LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet9/16, changed state to down
```

抽出用の `props.conf` のスタanzasは、以下のとおりです。

```
[syslog] EXTRACT-<port_flapping> =  
Interface\s(?<interface>(?(media)[^\d]+)(?(slot)\d+)\s\/\s(?<port>\d+)\s)\s,\schanged  
\sstate\ssto\s(?<port_status>up|down)
```

5つの異なるフィールドは、名前グループとして抽出されていますのでご注意ください。 `interface`、`media`、`slot`、`port`、`port_status`

次の2つの手順は、フィールド抽出には必要ありませんが、抽出したフィールドを使って、ポートがフラッピングしているイベントを探し、レポートする方法について説明しています。

タグを使って、`eventtypes.conf` にいくつかのイベントタイプを定義します。

```
[cisco_ios_port_down] search = "changed state to down" tags = cisco ios port check status report  
success down
```

```
[cisco_ios_port_up] search = "changed state to up" tags = cisco ios port check status report success  
up
```

最後に、上述の内容を結び、ポートフラッピングの検索および結果のレポートを行う保存済み検索(`savedsearches.conf`)を作成します。

```
[port flapping] search = eventtype=cisco_ios_port_down OR eventtype=cisco_ios_port_up
starthoursago=3 | stats count by interface,host,port_status | sort -count
```

サブトークンからフィールドを作成

フィールド値がトークンの一部である場合は、エントリーを `field.conf` に追加しなければなりません。例えば、フィールドの値が "123"で、イベントには"foo123"がある場合。

`props.conf` は上述の説明に従って設定します。その後で、以下のエントリーを `fields.conf` に追加します。

```
[<fieldname>]
INDEXED = False
INDEXED_VALUE = False
```

- `<fieldname>` にフィールドの名前を入力します。
  - ◆ 例えば、フィールド名に "url" と設定した場合は、`[url]` と入力します。
- `INDEXED` および `INDEXED_VALUE` に `false` を設定します。
  - ◆ これにより、インデックスのトークン以外の値を検索するよう Splunk に指定します。

特定のソース、ソースタイプ、ホストに対する検索時間フィールド抽出を無効にする

`props.conf` を編集して特定のソース、ソースタイプ、またはホストに対する検索時間フィールド抽出を無効にすることができます。`props.conf` の適切な `[<spec>]` に `KV_MODE = none` を追加します。

```
[<spec>]
KV_MODE = none
```

`<spec>` では以下が使えます。

- `<sourcetype>` はイベントのソースタイプ。
- `host::<host>`、`<host>` はイベントをホスト。
- `source::<source>`、`<source>` はイベントのソース。

## 検索時間フィールド抽出の管理

検索時間フィールド抽出の管理

管理のフィールド抽出ページを使って、Splunk Web のインタラクティブなフィールド抽出 (IFX) または `conf` ファイルの変更により作られた検索時間のフィールド抽出を管理します。フィールド抽出ページでは以下が行えます。

- Splunk のインスタンスにあるすべての Apps に対して作成した、または見る権限のある抽出の全セットを見直します。
- 抽出したフィールドに対する役割ベースの権限を更新します。これは、この抽出は、権限が更新されるまでは作成者しか使用することができないため、IFX によるフィールド抽出で重要です。
- `props.conf` に定義されたインラインランザクシヨンの正規表現を更新します。
- `transforms.conf` に定義された名前付き抽出を追加または削除します。
- 作成したまたは `書き込み` 権限のあるフィールド抽出を削除します。

**管理** > フィールド抽出の順に選択して、フィールド抽出ページを表示します。

管理で検索時間フィールド抽出をレビューする

`props.conf` および `transforms.conf` ファイルでフィールド抽出がどのように設定されているかを理解しておく、管理のフィールド抽出ページで抽出したフィールドを表示する方法を理解する役に立ちます。`props.conf` でフィールド抽出を定義する方法は、本書の「検索タイムのフィールド追加」で説明しています。

フィールド抽出は、`transforms.conf` の変換として設定できます。この設定方法については、管理者マニュアルの `transforms.conf` および `props.conf` ファイルの仕様を参照してください。

#### 名前カラム

フィールド抽出ページの名前カラムは、フィールド抽出の名前全体を `props.conf` に見られる形で表示します。その形式は以下のとおりです。

`<spec>` : [EXTRACT-`<class>` | REPORT-`<value>`]

- `<spec>` は以下が使えます。
  - ◆ `<sourcetype>`、イベントのソースタイプ。
  - ◆ `host::<host>`、`<host>` はイベントをホスト。
  - ◆ `source::<source>`、`<source>` はイベントのソース。

EXTRACT-`<class>` フィールド抽出は、`props.conf` に全体が定義された抽出です。これは、IFX および特定の検索コマンドで作成したフィールド抽出で自動生成されます。また、`props.conf` ファイルを直接更新して追加することもできます。この種の抽出は、抽出カラムに表示される正規表現と常に関連付けられています。

REPORT-`<value>` フィールド抽出は、正規表現が記述されている `transforms.conf` のスタンザにリンクされています。

#### タイプカラム

フィールド抽出の種類には、*inline* および *transforms.conf* の 2 種類があります。

- *Inline* 抽出は、通常 Splunk Web の IFX または検索コマンドを通じてインラインで定義されますが、設定ファイルを更新しても作成することができます。インライン抽出は、常に EXTRACT-`<class>` 名前設定を持ち、常に `props.conf` ファイルに定義されています。
- *Transforms.conf* 抽出は、`transforms.conf` および `props.conf` に手動で定義されます。*Transforms.conf* 抽出にも、常に REPORT-`<value>` 名前設定があります。

#### 表現カラム

表現カラムでは、管理がフィールド抽出タイプにより異なる内容を表示します。

- *inline* 抽出の場合、管理は Splunk がフィールドの抽出に使う正規表現を表示します。正規表現にある名前付きグループ(または複数グループ)は、抽出されるフィールドを示します。
- *transforms.conf* 抽出の場合、管理は、`props.conf` でフィールド抽出がリンクされる `transforms.conf` フィールド抽出スタンザ(または複数スタンザ)の名前を表示します。例えば、表現カラムに *access-extractions* と *ip-extractions* を抽出する 2 つの値を表示します。これは、`props.conf` に以下のように表示されます。

```
[access_combined] REPORT-access = access-extractions ip-extractions
```

この例では、`access-extractions` および `ip-extractions` の両方が、`transforms.conf` のフィールド抽出スタanzas の名前です。各スタanzas には、1 つ以上のフィールド抽出に使用される正規表現が含まれます。

#### フィールド抽出の更新

あらゆるフィールド抽出に対して、表現カラムに表示される値を編集できます。Splunk でそのフィールド抽出に対する詳細ページを開くため、編集するフィールド抽出の名前をクリックします。`inline` 抽出の正規表現を編集して、`transforms.conf` フィールド抽出のスタanzas 名を追加または削除できます。

**注記：** `Transforms.conf` フィールド抽出には、少なくとも 1 つの有効な `transforms.conf` フィールド抽出スタanzas 名を含んでいる必要があります。

#### フィールド抽出権限の更新

フィールド抽出をインライン法(IFX または検索コマンドなど)で作成した場合、そのフィールドは最初作成者しか使用できません。他のユーザーもフィールド抽出を使用できるようにするためには、その権限を更新する必要があります。そうするには、フィールド抽出ページでフィールド抽出を検索して、その権限リンクを選択します。これにより、知識オブジェクト(保存済み検索、イベントタイプ、検索マクロ、ナビゲーションメニューなど)に対する管理者が使用する標準の権限管理ページが表示されます。

このページでは、フィールド抽出に対する役割ベースの権限を設定し、それが特定の App のユーザーに利用可能かどうか、またはすべての App のユーザーに利用可能かどうかなどを指定できます。

#### フィールド抽出の削除

管理のフィールド抽出ページでは、その権限を持つ限り、フィールド抽出を削除できます。削除するフィールド抽出に対して削除をクリックします。

# インデックスタイムフィールド抽出のカスタマイズ

## インデックスタイムフィールド抽出のカスタマイズ

Splunk がインデックスタイムで抽出およびインデックスする一連のデフォルトフィールド (timestamp、punct、host、source、sourcetype など) はカスタマイズしないでください。このフィールド一覧に追加すると、インデックスされた各フィールドで検索可能なフィールドのサイズが増大するため、インデックスの性能および検索タイムに悪影響を及ぼします。デフォルトフィールドも、その一覧に変更を加えるなどの操作を行うと、データセット全体を再インデックスする必要があります。

これらの注意事項をふまえて、デフォルトフィールドを変更または追加する必要がある場合に直面することがあります。例えば、特定の検索時間のフィールド抽出で、検索性能に明らかに影響を及ぼしている場合があります。これは、例えば、`foo!=bar` または `or NOT foo=bar` などの表現で大規模なイベントを共有検索し、`foo` フィールドが `bar` の値を参照するときほぼ常に発生します。

反面、検索時間で抽出された値がフィールドの外側にまれに存在する場合などデフォルトフィールドを更新したい場合があります。例えば、通常 `foo=1` のみに対して検索を行うと、`foo=1` を持たない多くのイベントで `1` が発生する場合があります。Splunk のインデックスタイムで抽出されるデフォルトフィールドの一覧に `foo` を追加できます。

## 追加デフォルトフィールドの定義

`props.conf`、`transforms.conf`、`fields.conf` を編集して追加のデフォルトフィールドを定義します。

`$SPLUNK_HOME/etc/system/local/`、または `$SPLUNK_HOME/etc/apps/` の独自のカスタムアプリケーションディレクトリにあるファイルを編集します。設定ファイルの全般的な内容については、管理者マニュアルの「設定ファイルについて」を参照してください。

Splunk で指定できるフィールド名は、以下のアルファベット文字またはアンダーラインのみです。

- フィールド名に指定できる文字：a-z, A-Z, 0-9, \_
- フィールド名の最初の文字に 0-9 または \_ は指定できません。アンダーライン(\_)から始まる名前は、Splunk の内部変数に使用されています。
- 国際文字は使用できません。

## `transforms.conf` へ新しいデフォルトフィールドに対する正規表現の追加

`transforms.conf` に以下の行を追加します。

```
[<unique_stanza_name>]
REGEX = <your_regex>
FORMAT = <your_custom_field_name>::"$1"
WRITE_META = true
```

- `<unique_stanza_name>` でスタンザの名前を付けます。この名前を後で使って `props.conf` を設定します。
- `REGEX =` は、カスタムフィールド値を識別する正規表現を作ります。

- `FORMAT = 正規表現で$1` として抽出した値の前に `<your_custom_field_name>` を挿入します。
  - ◆ Splunk Web で空白を含むフィールド値を正しく表示するためには、`FORMAT` キーに引用符を適用します。
  - ◆ `FORMAT = <your_custom_field_name>::"$1"`
  - ◆ 複数のグループと一致する 1 つの正規表現を使って複数フィールドを抽出できます。  
`FORMAT = <your_first_field>::"$1" <your_second_field>::"$2"`
- `WRITE_META =` ここで、フィールド名を書き込むよう `true`、値に Splunk がデフォルトフィールドを格納する `_meta` と設定します。(下の「Splunk でデフォルトフィールドを作成する方法」を参照してください。)

**注記:** 正規表現で読み込むグループは、ASCII 文字を使うフィールド名((a-zA-Z0-9\_))を特定する必要があります。国際文字は機能しません。

新しいデフォルトフィールドを `props.conf` にリンク

`props.conf` に以下の行を追加します。

```
[<spec>]
TRANSFORMS-<value> = <unique_stanza_name>
```

- `<spec>` は以下が使えます。
  - ◆ `<sourcetype>`、イベントのソースタイプ。
  - ◆ `host=<host>`、`<host>` はイベントに対するホスト。
  - ◆ `source=<source>`、`<source>` はイベントに対するソース。
- `<unique_stanza_name>` は、`transforms.conf` のスタンザの名前。
- `<value>` は任意の値です。名前空間に属性を与えます。

**注記:** インデックスタイムのフィールド抽出の場合、`props.conf` は、`EXTRACT-<value>` ではなく `TRANSFORMS-<class>` を検索時間のフィールド抽出の設定に使用します。

新しいデフォルトフィールドに対する `fields.conf` にエントリーを追加

新しいインデックスフィールドに対する `fields.conf` に以下のエントリーを追加します。

```
[<your_custom_field_name>]
INDEXED=true
```

- `<your_custom_field_name>` は、`transforms.conf` に追加した固有のスタンザに設定するカスタムフィールドの名前。
- `INDEXED=true` を設定して、フィールドがインデックスされたことを示します。

**注記:** 検索時間で同じ名前のフィールドが抽出された場合は、フィールドに `INDEXED=false` を設定しなければなりません。さらに、そのフィールドの値を持つイベントがインデックスタイムで抽出されず、検索時間で抽出された場合も、`INDEXED_VALUE=false` を設定する必要があります。

例えば、インデックスタイムで単純な `<field>::1234` 抽出を実施するとします。これは機能しますが、`A(¥d+)B` などの正規表現を基に検索時間のフィールド抽出を実施した場合、`A1234B` という文字列から `1234` というフィールド値が生成されるという問題が発生することがあります。これは、Splunk がインデックスタイムで `<field>::1234` の抽出を探ることができず、検

検索時間で 1234 に対するイベントを返す場合があります。

**Splunk** を再起動して変更を有効にする

`props.conf` および `transforms.conf` などの設定ファイルへの変更は、**Splunk** を終了して再起動するまで適用されません。

**Splunk** でデフォルトフィールドを作成する方法

**Splunk** は、`_meta` に記述してインデックスフィールドを作成します。その手順は以下のとおりです。

- `_meta` は、`DEST_KEY = _meta` または `WRITE_META = true` のいずれかを含む `transforms.conf` で一致するすべての変換により変更されます。
- それぞれの一致する変換は、`_meta` を上書きするので、`WRITE_META = true` を使って `_meta` を追加します。
  - ◆ `WRITE_META` を使わない場合は、`FORMAT` を `$0` で開始します。
- 構文解析中に `_meta` を完全に作成した後は、**Splunk** が次の方法でテキストを解釈します。
  - ◆ テキストは、ユニットに分割されます。ユニットは空白で区別されます。
  - ◆ 引用符 (" ") は、空白に関係なく文字をグループ化して大きなユニットにまとめます。
  - ◆ 引用符直前にあるバックスラッシュ ( \ ) は、引用符のグループ化特性を無効にします。
  - ◆ バックスラッシュの前に付くバックスラッシュはそのバックスラッシュを無効にします。
  - ◆ ダブルコロンの (::) を含むテキストは、抽出されたフィールドに変わります。ダブルコロンの左側のテキストは、フィールド名となり、右側は値となります。

**注記：** 正規表現で抽出された値を持つインデックスフィールドに引用符が付いている場合は、通常、機能しません。また、バックスラッシュが問題となる場合があります。検索時間で抽出されたフィールドにはこのような制限はありません。

ここに、引用符およびバックスラッシュを無効にするための引用符およびバックスラッシュを含む一連のインデックスタイム抽出の例を紹介します。

```
WRITE_META = true
FORMAT = field1::value field2::"value 2" field3::"a field with a \" quotation mark" field4::"a field which ends with a backslash\"
```

**Splunk** でフィールド名を作成する場合

**Splunk** でフィールド名を作るとき、インデックスタイムまたは検索時間による抽出に関わらず、すべての抽出フィールドに対してデフォルトまたはカスタム設定で以下の規則を適用しています。

- a-z、A-Z、0-9 の範囲外のすべての文字は、アンダーライン ( \_ ) に置き換えられます。
- 文頭のアンダーラインはすべて削除されます (**Splunk** では、アンダーラインで始まるフィールドは内部変数に使用しません)。

## 検索時間フィールド抽出例

インデックスタイムのデフォルトフィールド抽出に対する設定ファイルの設定例を以下に示します。

## 新しいデフォルトフィールドの定義

この例では、`err_code` と呼ばれるデフォルトフィールドを作成します。

### transforms.conf

`transforms.conf` に以下を追加します。

```
[netscreen-error]
REGEX = device_id=%[w+%] (?<err_code>[^:]+)
FORMAT = err_code::"$1"
WRITE_META = true
```

このスタンプは、`device_id=` の後に括弧付きの文字を記述し、コロンでテキスト文字列を終了します。イベントのソースタイプは、`testlog` です。

コメント：

- `FORMAT =` 行には以下の値が含まれます。
  - ◆ `err_code::` はフィールドの名前。
  - ◆ `$1` はインデックスに記述される新しいフィールドを指す。これは `REGEX` で抽出された値。
- `WRITE_META = true` は、インデックスに `FORMAT` のコンテンツを書き込む指示。

### props.conf

`props.conf` に以下の行を追加します。

```
[testlog]
TRANSFORMS-netscreen = netscreen-error
```

### fields.conf

`fields.conf` に以下の行を追加します。

```
[err_code]
INDEXED=true
```

## 1つの正規表現で新しいデフォルトフィールドを定義

この例では、`username` と `login_result` 呼ばれる2つのインデックスフィールドを作成します。

### transforms.conf

`transforms.conf` に以下を追加します。

```
[ftpd-login]
REGEX = Attempt to login by user: (.*) : login (.*)\
FORMAT = username::"$1" login_result::"$2"
WRITE_META = true
```

このスタンプは、文字テキスト `Attempt to login by user:` を検索し、コロンに続いてユーザー名を抽出し、結果の後にピリオドを表示します。結果は以下のとおりです。

```
2008-10-30 14:15:21 mightyhost awesomeftpd INFO Attempt to login by user: root: login
FAILED.
```

#### props.conf

`props.conf` に以下の行を追加します。

```
[ftpd-log]
TRANSFORMS-login = ftpd-login
```

#### fields.conf

`fields.conf` に以下の行を追加します。

```
[username]
INDEXED=true

[login_result]
INDEXED=true
```

## 外部データソースのフィールド検索

### 外部データソースのフィールド検索

ダイナミックなフィールド検索機能を使って、静的表(CSV ファイル)または外部(Python)コマンドなど、外部ソースの情報を持つイベントにフィールドを追加します。また、時間情報でより高度な検索をすることができます。

例えば、Splunk のログインをモニタリングしていて、Splunk のインデックスにアクセスの IP アドレスとタイムスタンプを持つ場合、ダイナミックなフィールド検索を使って、IP アドレスとタイムスタンプを、DHCP ログにある IP およびタイムスタンプデータと一致する MAC アドレスとユーザー名情報にマップすることができます。

### 検索の設定手順

1. `transforms.conf` を編集して検索テーブルを定義します。

現在は、静的検索(CSV ファイルを使用)と外部検索(スクリプトを使用)の 2 種類の検索テーブルが定義できます。変換スタンプで使用する引数は、定義する検索テーブルの種類を示します。静的検索には `filename`、外部検索には `external_cmd` を使用します。

**注記:** 1 つの検索テーブルには、2 つ以上のカラムが必要です。各カラムには、同じ値を持つ複数のインスタンスを持つことができます。(マルチバリューフィールド)

2. `props.conf` を編集して検索テーブルを適用します。

このステップは、静的検索および外部検索と同じです。この設定ファイルでは、フィールドに `transforms.conf` で定義した検索テーブルの一致および出力を指定します。

3. Splunk を再起動して設定ファイルへの変更を有効にします。

再起動が完了すると、フィールドの選択に一覧される検索テーブルに出力フィールドが表示されます。ここから、一致する各イベントに対して表示するフィールドが選択できます。

**重要：** `$SPLUNK_HOME/etc/system/default` の `conf` ファイルは編集しないでください。代わりに、`$SPLUNK_HOME/etc/system/local/` または `$SPLUNK_HOME/etc/apps/<app_name>/local/` のファイルを編集します。

静的ファイルを基にしたフィールド検索の設定

最も簡単なフィールド検索は、静的テーブル(CSV ファイル)を基に作成します。CSV ファイルは、必ず以下のいずれかの場所に保存します。

- `$SPLUNK_HOME/etc/system/lookups/`
- `$SPLUNK_HOME/etc/apps/<app_name>/lookups/`

**重要：** この検索ディレクトリが存在しない場合は、必ず作成してください。

1. `transforms.conf` を編集して検索テーブルを定義します。

`transforms.conf` で、検索テーブルを定義するスタンザを追加します。スタンザの名前は、検索テーブルの名前です。この変換は `props.conf` で使用します。

このスタンザでは、CSV ファイルの名前を参照します。

```
[myLookup]
filename = <filename>
max_matches = <integer>
```

任意で、イベントに適用する一致エントリーの数を指定できます。`max_matches` は、最初(最初のファイル)の `<integer>` エントリーが使用されることを示します。デフォルトでは、`max_matches` は時間ベースではない検索に対して 1000 と設定されています。

2. `props.conf` を編集して検索テーブルを適用します。

`props.conf` で、`lookup` キーを持つスタンザを追加します。このスタンザは、`transforms.conf` で定義した検索テーブルを指定し、Splunk がイベントに適用する方法を示します。

```
[<stanza name>]
lookup_<class> = $TRANSFORM <match_field_in_table> OUTPUT <output_field_in_table>
```

- `$TRANSFORM` は、検索テーブルを定義した `transforms.conf` のスタンザを参照します。
- `match_field_in_table` は、値一致に使う検索テーブルのカラムです。

- output\_field\_in\_table は、イベントに追加した検索テーブルのカラムです。
- 検索のどちら側にも複数のカラムを持つことができます。例えば、\$TRANSFORM <match\_field1>、<match\_field2> OUTPUT <match\_field3>、<match\_field4>を持つことができます。1つのフィールドから2つのフィールド、3つのフィールドから1つのフィールドなどに返すように設定することができます。

検索テーブルのフィールド名とイベントが一致しない場合、またはイベントのフィールドの名前を変更したい場合は、AS 節を使います。

```
[<stanza name>]
lookup_<class> = $TRANSFORM <match_field_in_table> AS <match_field_in_event>
OUTPUT <output_field_in_table> AS <output_field_in_event>
```

OUTPUT 節の後には複数のフィールドを指定できます。OUTPUT を使用しない場合は、Splunk が検索テーブルからすべてのフィールド名と値をイベントに追加します。

### 3. Splunk を再起動します。

#### 静的フィールド検索の例

access\_combined ログの HTTP ステータスコードに対する検索の設定例をここに示します。この例では、検索テーブル (http\_status.csv) の status フィールドとイベントのフィールドを一致させます。その後、ステータスの説明とステータスの種類をイベントに追加します。

以下は http\_status.csv ファイルの内容です。これを、\$SPLUNK\_HOME/etc/apps/<app\_name>/lookups/ に保存します。これを検索 App で使用する場合は、ファイルを \$SPLUNK\_HOME/etc/apps/search/lookups/ に保存します。

```
status,status_description,status_type
100,Continue,Informational
101,Switching Protocols,Informational
200,OK,Successful
201,Created,Successful
202,Accepted,Successful
203,Non-Authoritative Information,Successful
204,No Content,Successful
205,Reset Content,Successful
206,Partial Content,Successful
300,Multiple Choices,Redirection
301,Moved Permanently,Redirection
302,Found,Redirection
303,See Other,Redirection
304,Not Modified,Redirection
305,Use Proxy,Redirection
307,Temporary Redirect,Redirection
400,Bad Request,Client Error
401,Unauthorized,Client Error
402,Payment Required,Client Error
403,Forbidden,Client Error
404,Not Found,Client Error
405,Method Not Allowed,Client Error
406,Not Acceptable,Client Error
407,Proxy Authentication Required,Client Error
408,Request Timeout,Client Error
409,Conflict,Client Error
410,Gone,Client Error
```

411,Length Required,Client Error  
412,Precondition Failed,Client Error  
413,Request Entity Too Large,Client Error  
414,Request-URI Too Long,Client Error  
415,Unsupported Media Type,Client Error  
416,Requested Range Not Satisfiable,Client Error  
417,Expectation Failed,Client Error  
500,Internal Server Error,Server Error  
501,Not Implemented,Server Error  
502,Bad Gateway,Server Error  
503,Service Unavailable,Server Error  
504,Gateway Timeout,Server Error  
505,HTTP Version Not Supported,Server Error

1. `$(SPLUNK_HOME)/etc/system/local/` または `$(SPLUNK_HOME)/etc/apps/<app_name>/local/` のいずれかにある `transforms.conf` ファイルに以下を記述します。

```
[http_status]
filename = http_status.csv
```

2. `$(SPLUNK_HOME)/etc/system/local/` または `$(SPLUNK_HOME)/etc/apps/<app_name>/local/` のいずれかにある `props.conf` ファイルに以下を記述します。

```
[access_combined]
lookup_table = http_status status OUTPUT status_description, status_type
```

3. Splunk を再起動します。

#### 検索結果を使用した検索テーブルの設定

保存済み検索の結果を使って検索テーブルを設定できます。ローカルまたはアプリケーション専用の `savedsearches.conf` で、以下を行います。

1. 検索を定義します。任意で、検索検索コマンドで使用する検索をテストして正しいことを確認します。
2. 検索による入力操作を有効にします。
3. Splunk に検索テーブルをコピーする場所を指示します。ステップ 2 および 3 で、保存済み検索に対するスタンザに以下の 2 行を追加します。

```
action.populate_lookup = 1
action.populate_lookup.dest = <string>
```

`action.populate_lookup.dest` の値は、Splunk が検索結果を書き込む CSV ファイルへのパスです。この操作が機能するためには、予め保存先のディレクトリが存在している必要があります。このディレクトリには、

`$(SPLUNK_HOME)/etc/system/lookups` または `$(SPLUNK_HOME)/etc/<app_name>/lookups` のいずれかを使用します。

Splunk は保存済み検索の結果を CSV ファイルにコピーするため、フィールド検索を静的検索の設定と同じ方法で設定することができます。

## 外部コマンドを基にしたフィールド検索の設定

外部検索の場合、`transforms.conf` のスタanzasは、コマンドまたはスクリプトと引数を参照して呼び出します。また、呼び出すコマンドまたはスクリプトの種類を指定することもできます。

```
[myLookup]
external_cmd = <string>
external_type = python fields_list =
<string> max_matches = <integer>
```

`fields_list` を使い、外部コマンドが対応するコンマとスペースで区切られたすべてのフィールドを一覧します。

**注記：** 現在、Splunk は、外部コマンドベースのフィールド検索に Python スクリプトのみをサポートしています。これらの検索に使用される Python スクリプトは、必ず次のいずれかに保存しなければなりません。

- `$SPLUNK_HOME/etc/apps/<app_name>/bin`
- `$SPLUNK_HOME/etc/searchscripts`

## 外部フィールド検索の例

外部検索を使って、DNS サーバーの情報と一致させる方法の例をここに示します。この例では、`dnslookup.py` が以下を行うスクリプトです。

ホストが与えられている場合は、IP アドレスを返す

IP が与えられている場合は、ホスト名を返す

1. `transforms.conf` ファイルに、以下を記述します。

```
[dnsLookup]
external_cmd = dnslookup.py host ip
fields_list = host, ip
```

2. `props.conf` ファイルに、以下を記述します。

```
[access_combined]
lookup_dns = dnsLookup host OUTPUT ip
```

DNS 逆引きの場合は、`props.conf` stanzasは以下ようになります。

```
[access_combined]
lookup_rdns = dnsLookup ip OUTPUT host
```

3. Splunk を再起動します。

## 時間ベースのフィールド検索の設定

静的または外部検索テーブルに時間を表すフィールド値が含まれている場合、この時間フィールドを使ってフィールド検索を設定できます。時間ベースの検索では、以下の行を `transforms.conf` の検索スタンプに追加します。

```
time_field = <field_name>
time_format = <string>
```

`time_field` が存在する場合は、デフォルトで `max_matches` に 1 が設定されます。また、降順で最初に一致したエントリーが適用されます。

`time_format` キーを使って `time_field` の `strptime` フォーマットを指定します。デフォルトの `time_format` は UTC です。

時間ベースの検索で一致する場合、イベントが検索のエントリーより遅い場合に備えて時間量の最大および最小のオフセットを指定できます。これは、スタンプに以下の行を追加して実施します。

```
max_offset_secs = <integer>
min_offset_secs = <integer>
```

デフォルトでは、最大オフセットはなく、最小オフセットには 0 が設定されています。

## 時間ベースのフィールド検索の例

IP アドレスとタイムスタンプを基に DHCP ログを使ってネットワークのユーザーを特定する方法例をここに示します。DHCP ログがファイル (`dhcp.csv`) に存在し、タイムスタンプ、IP アドレス、ユーザー名、MAC アドレスが含まれていると仮定します。

1. `transforms.conf` ファイルに、以下を記述します。

```
[dhcpLookup]
filename = dhcp.csv
time_field = timestamp
time_format = %d/%m/%y %H:%M:%S
```

2. `props.conf` ファイルに、以下を記述します。

```
[dhcp]
lookup_table = dhcpLookup ip mac OUTPUT user
```

3. Splunk を再起動します。

## ソース入力時にファイルヘッダーからフィールドを抽出

### ソース入力時にファイルヘッダーからフィールドを抽出

CSV ファイルや MS Exchange のログファイルなど、特定のデータソースとソースタイプには、フィールド情報を含むヘッダーを持つことができます。Splunk で、これらのフィールドをソース入力時に自動抽出するよう設定できます。

例えば、基本的に静的なテーブル形式である従来の CSV ファイルは、以下のようなヘッダー行を持つことができます。

```
name, location, message, "start date"
```

これは、ファイル内で後述される値に対する一連のカラムヘッダーと同様に機能します。

**注記：** ヘッダーベースのフィールド自動抽出は、ソース入力時(インデックスタイムの前)に行われるため、インデックスのサイズや性能に悪影響を及ぼしません。

#### ヘッダーベースのフィールド自動抽出のしくみ

特定のソースまたはソースタイプに対するヘッダーベースのフィールド自動抽出の場合、Splunk はヘッダーフィールド情報をスキャンして、その後フィールド抽出に使用します。ソースに必要なヘッダー情報がある場合、Splunk は、区切り文字ベースのキー/値抽出を使ってフィールドを抽出します。

Splunk は、そのソースの `transforms.conf` にエントリーを作成して、フィールドを抽出するための変換を行って値を入力します。また、Splunk は、ソースタイプスタanzas を `props.conf` に追加して、フィールド抽出変換とソースを関連付けます。その後、Splunk は、検索時間にソースからのイベントに変換を適用します。

検索ビューで別のフィールドをフィールドサイドバーから選択するのと同じように ( **フィールドの選択** を選択して利用可能なすべてのフィールドの一覧を参照 ), Splunk により抽出されたフィールドを使って、フィールドを絞込みおよびレポートできます。

#### ヘッダーベースのフィールド自動抽出を有効にする

`props.conf` を編集して任意のソースまたはソースタイプに対してヘッダーベースのフィールド自動抽出を有効にします。  
`$SPLUNK_HOME/etc/system/local/` または `$SPLUNK_HOME/etc/apps/` の独自のカスタムアプリケーションディレクトリにあるこのファイルを編集します。

設定ファイルの全般的な内容については、管理者マニュアルの「設定ファイルについて」を参照してください。

ソースまたはソースタイプに対するヘッダーベースのフィールド自動抽出を実行するには、`props.conf` のそのソースまたはソースタイプのスタanzas の下に `CHECK_FOR_HEADER=TRUE` を追加します。

**重要：** ヘッダーベースのフィールド自動抽出を有効にしたいソースに対するソースタイプを既に定義してある場合は、

`props.conf` で `CHECK_FOR_HEADER=TRUE` を設定する **前に**、`inputs.conf` のスタanzas を編集して `sourcetype = [name]` を削除し、自動抽出で生成される値が衝突しないようにする必要があります。

#### MS Exchange ソースに対する `props.conf` エントリーの例

```
[MSExchange]
CHECK_FOR_HEADER=TRUE
...
```

**注記：** `CHECK_FOR_HEADER=FALSE` を設定して、ソースまたはソースタイプに対するヘッダーベースのフィールド自動抽出をオフにします。

**重要：** `props.conf` で行った変更 (ヘッダーベースのフィールド自動抽出の有効化など) は、Splunk を再起動するまで有効になりません。

#### Splunk により行われる設定ファイルの変更

ソースまたはソースタイプに対するヘッダーベースのフィールド自動抽出を有にすると、Splunk は、そのソースまたはソースタイプに対するフィールドを抽出する際に、`SPLUNK_HOME/etc/apps/learned/` の `transforms.conf` および `props.conf` のコピーにスタanzasを追加します。

**重要：** Splunk が追加した後でスタanzasを編集しないでください。関連する抽出フィールドが機能しなくなります。

Splunk は、固有のヘッダー情報が `props.conf` に定義されたソースタイプと一致する各ソースタイプの `transforms.conf` にスタanzasを作成します。Splunk は、各スタanzasに `[AutoHeader-M]` の形式で名前を付けます。このとき、`M` は固有のヘッダーを持つ各ソースに対して順次に増加する整数です (例: `[AutoHeader-1]`、`[AutoHeader-2]`、...、`[AutoHeader-M]`)。Splunk は、そのフィールドを変換 (ヘッダー情報を使う) して各スタanzasに値を入力します。

**重要：** ヘッダーベースのフィールド自動抽出を有効にしたいソースに対するソースタイプを既に定義してある場合は、`props.conf` で `CHECK_FOR_HEADER=TRUE` を設定する前に、`inputs.conf` のスタanzasを編集して `sourcetype = [name]` を削除し、自動抽出で生成される値が衝突しないようにする必要があります。

前述の例でヘッダーベースのフィールド自動抽出が有効にされている MS Exchange ソースに対して、Splunk が自動生成する `transforms.conf` エントリーの例をここに示します。

```
...
[AutoHeader-1]
FIELDS="time", "client-ip", "cs-method", "sc-status"
DELIMS=" "
...
```

Splunk はその後、それぞれの固有ソースに対して新しいソースタイプのスタanzasを `props.conf` に追加します。Splunk は、そのスタanzasに `[yoursource-N]` の形式で名前を付けます。このとき、`yoursource` は、ヘッダーベースのフィールド自動抽出で設定されたソースタイプであり、`N` は、`transforms.conf` の各変換に対応して順次増加する整数です。

`props.conf` エントリーの例(説明された MS Exchange ファイルを含む)

```
# the original source you configured
[MSExchange] CHECK_FOR_HEADER=TRUE
...
# source type that Splunk added to <code>transforms.conf</code> to handle transforms for automatic
header-based field extraction for the same source
[MSExchange-1]
REPORT-AutoHeader = AutoHeader-1
...
```

## 検索およびヘッダーベースのフィールド抽出に関する注意事項

ワイルドカードを使って、Splunk がヘッダーベースのフィールド抽出で生成したソースタイプに関連するイベントを検索します。

例えば、`sourcetype="yoursource"` の検索は以下のようになります。

```
sourcetype=yoursource*
```

## ヘッダーベースのフィールド自動抽出の例

この例では、ヘッダーベースのフィールド抽出が一般的なソースタイプを取扱うしくみについて説明します。

### MS Exchange ソースファイル

この例では、ヘッダーベースのフィールド自動抽出を使って、MS Exchange ファイルからフィールドを抽出する方法について説明します。

このサンプルでは、MS Exchange ログファイルのヘッダーにスペースで区切られたフィールド名の一覧が含まれています。

```
# Message Tracking Log File
# Exchange System Attendant Version 6.5.7638.1
# Fields: time client-ip cs-method sc-status
14:13:11 10.1.1.9 HELO 250
14:13:13 10.1.1.9 MAIL 250
14:13:19 10.1.1.9 RCPT 250
14:13:29 10.1.1.9 DATA 250
14:13:31 10.1.1.9 QUIT 240
```

Splunk は `transforms.conf` にヘッダーおよび変換を以下のように作成します。

```
[AutoHeader-1]
FIELDS="time", "client-ip", "cs-method", "sc-status"
DELIMS=" "
```

Splunk は自動的に区切り文字として空白を検出することに注意してください。

その後 Splunk は、これを、`props.conf` のソースタイプスタanzaに追加して変換とソースを関連付けます。

```
# Original source type stanza you create
[MSExchange]
CHECK_FOR_HEADER=TRUE
...
# source type stanza that Splunk creates
[MSExchange-1]
REPORT-AutoHeader = AutoHeader-1
...
```

Splunk は、各イベントから以下のフィールドを自動抽出します。

```
14:13:11 10.1.1.9 HELO 250
    • • time="14:13:11" client-ip="10.1.1.9" cs-method="HELO" sc-status="250"
14:13:13 10.1.1.9 MAIL 250
    • • time="14:13:13" client-ip="10.1.1.9" cs-method="MAIL" sc-status="250"
```

```
14:13:19 10.1.1.9 RCPT 250
```

- • time="14:13:19" client-ip="10.1.1.9" cs-method="RCPT" sc-status="250"

```
14:13:29 10.1.1.9 DATA 250
```

- • time="14:13:29" client-ip="10.1.1.9" cs-method="DATA" sc-status="250"

```
14:13:31 10.1.1.9 QUIT 240
```

- • time="14:13:31" client-ip="10.1.1.9" cs-method="QUIT" sc-status="240"

## CSV ファイル

この例では、ヘッダーベースのフィールド自動抽出を使って CSV ファイルからフィールドを抽出する方法について説明します。

### CSV ファイルの例

```
foo,bar,anotherfoo,anotherbar
100,21,this is a long file,nomore
200,22,wow,o rly?
300,12,ya rly!,no wai!
```

Splunk は `transforms.conf` (`$SPLUNK_HOME/etc/apps/learned/transforms.conf` に保存されている) にヘッダーおよび変換を以下のように作成します。

```
# Some previous automatic header-based field extraction
[AutoHeader-1]
...
# source type stanza that Splunk creates
[AutoHeader-2]
FIELDS="foo", "bar", "anotherfoo", "anotherbar"
DELIMS=","
```

Splunk は自動的に区切り文字としてコンマを検出することに注意してください。

その後 Splunk は、これを、`props.conf` の新しいソースタイプスタanzaに追加して変換とソースを関連付けます。

```
...
[CSV-1]
REPORT-AutoHeader = AutoHeader-2
...
```

Splunk は、各イベントから以下のフィールドを抽出します。

```
100,21,this is a long file,nomore
• • foo="100" bar="21" anotherfoo="this is a long file" anotherbar="nomore"
200,22,wow,o rly?
• • foo="200" bar="22" anotherfoo="wow" anotherbar="o rly?"
300,12,ya rly!,no wai!
• • foo="300" bar="12" anotherfoo="ya rly!" anotherbar="no wai!"
```

# 複数の値を持つフィールドの設定

## 複数の値を持つフィールドの設定

`fields.conf` にマルチバリューフィールドを設定して、1 つ以上のフィールド値を 1 つの抽出されたフィールド値で認識する方法を Splunk に指示します。 `$SPLUNK_HOME/etc/system/local/`、または `$SPLUNK_HOME/etc/apps/` の独自のカスタムアプリケーションディレクトリにある `fields.conf` を編集します。

設定ファイルの一般的な内容については、管理者マニュアルの「設定ファイルについて」を参照してください。

Splunk は、検索時にマルチバリューフィールドを構文解析し、検索パイプラインでその値を処理できるようにします。マルチバリューフィールドを使って作業できる検索コマンドは、`makemv`、`mvcombine`、`mvexpand`、`nomv` などです。これらを含むコマンドの詳細については、検索リファレンスを参照してください。

## `fields.conf` による複数の値を持つフィールドの設定

マルチバリューフィールドのスタanzas を `fields.conf` に追加してマルチバリューフィールドを定義します。 `tokenizer` キーを持つ正規表現を定義することによりフィールド値から値を構文解析する方法を Splunk に指示します。

**注記：** フィールドを設定する他の属性がある場合、`tokenizer` の下の同じスタanzas に設定します。 詳しくは、管理者マニュアルの `fields.conf` に関する説明を参照してください。

```
[<field name>]
tokenizer = $REGEX
```

- ここに `props.conf` および `transforms.conf` で定義したフィールドの名前を設定します。
- フィールドはインデックスタイムまたは検索時間で抽出されます。
- `tokenizer` の場合、Splunk にフィールドをマルチバリューに構文解析する方法を伝える正規表現を定義します。

## 例

以下は、`$SPLUNK_HOME/etc/system/README/fields.conf.example` の例であり、電子メールを `To`、`From`、`CC` のマルチバリューに分割します。

```
[To]
TOKENIZER = (\w[\w.\-]*@[\w.\-]*\w)
[From]
TOKENIZER = (\w[\w.\-]*@[\w.\-]*\w)
[Cc]
TOKENIZER = (\w[\w.\-]*@[\w.\-]*\w)
```

# ホストの取り扱い

## ホストについて

### ホストについて

イベントの `host` 値は、イベントが発生したネットワーク上に存在する物理的なデバイスの名前です。 `host` フィールドを使って、特定のデバイスから生成されるすべてのデータを検索します。ホストにタグを付けて、共有の機能や設定を持つホストのグループからデータを検索します。 `Host` には、IP アドレス、ホスト名、完全修飾ドメイン名などがあります。 `Host` は、デフォルトフィールド、つまり、Splunk が各イベントのインデックスに `host` 値を割り当てます。

### Splunk でホスト値を割り当てる方法

ソースに対して別のホストルールが指定されていない場合、Splunk は `host` を特定の Splunk サーバーに入力されるすべてのデータに適用するデフォルト値に割り当てます。デフォルトのホスト値は、ネットワークホストのホスト名または IP アドレスです。Splunk をイベントが発生したサーバー上で起動する場合 (通常の稼働)、これが正しく、手動による設定は必要ありません。

Splunk サーバーに対するデフォルトホストを設定する方法を学びます。

### リモートアーカイブファイルに対するホストの上書き

中央ログアーカイブで Splunk を実行する、または同一環境の別のホストからコピーされたファイル进行处理する場合、特定の入力によるイベントに対するデフォルトのホスト割り当てを上書きする必要があります。入力のホスト割り当ての設定には 2 つの方法があります。その入力によるすべてのデータに対するカスタムホスト値を定義できます。また、割り当てたホスト値をソースのパスまたはファイル名の一部と一致させることができます。後者の方法は、各ホストのログアーカイブを異なるサブディレクトリに分離するディレクトリ構造がある場合に便利です。

### 中央のログサーバー環境から異なるホストを取得

複数のサーバーが関与する場合、中央のログホストが Splunk にイベントを送ります。中央のログサーバーは、*レポートホスト*と呼ばれています。イベントが発生したシステムは、*元となるホスト*(または *ホスト*)と呼ばれます。このような場合、中央のログホストから受信したイベントに対する自動ホスト割り当てを上書きするルールを定義する必要があります。

### ホスト値にタグを付ける

ホスト値にタグを付けると、検索の実行を向上させることができます。タグにより、ホストのグループを便利で検索可能なカテゴリにまとめることができます。

## inputs.conf のホスト値の設定

host 値を直接 inputs.conf に設定します。ホストによっては、transforms.conf および props.conf の抽出設定を変更する必要があります。設定ファイルを手動で変更する前には、設定ファイルについて知っておく必要があります。

# デフォルトの Splunk サーバーホストの設定

## デフォルトの Splunk サーバーホストの設定

イベントの host 値は、イベントが発生したネットワーク上に存在する物理的なデバイスの名前です。Splunk は、各イベントにインデックスを付けるインデックスタイムでホスト値を割り当てるため、ホスト値を検索すると、特定のデバイスで発生したすべてのデータを簡単に検索できます。

## デフォルトホストの割り当て

ソースに対して他のホストルールを指定していない場合(この情報および本書の別の節を使って)、イベントに対するデフォルトのホスト値は、通常、イベントが発生したネットワークホストのホスト名、IP アドレス、または完全修飾ドメイン名です。Splunk を実行するサーバーでイベントが発生する(最も代表的な状態)と、上述のホスト割り当てが行われ、ユーザーは何も変更する必要はありません。ただし、データが別のホストから転送されている場合、またはアーカイブデータを一括ロードする場合は、そのデータに対応するデフォルトホスト値に変更する場合があります。

ここでは、特定のデバイスで発生したイベントデータに対してデフォルトのホスト値を設定する方法について説明します。

## 管理を使ったデフォルトホスト値の設定

管理を使ってデフォルトのホスト値を設定します。

1. Splunk Web で、右上隅の**管理**リンクをクリックします。
2. **システム設定**をクリックします。
3. **インデックス設定**セクションの**デフォルトホスト名**値を変更します。

これで、別のホスト名を受信しないすべてのイベントに対するホストフィールドの値を設定します。

## 設定ファイルを使ったデフォルトホスト値の設定

このホスト割り当ては、Splunk のインストール時に inputs.conf に記述されます。\$SPLUNK\_HOME/etc/system/local/、または \$SPLUNK\_HOME/etc/apps/ の独自のカスタムアプリケーションディレクトリを編集してホストエントリを変更します。(カスタマイズしたデータを別のサーバーに簡単に転送したい場合は、後者を使用してください。)

inputs.conf のホスト割り当ては以下の形式で指定します。

```
host = <string>
```

- <string> をユーザーが選択したデフォルトのホスト値に設定します。<string> は、データが生成されたホストの IP アドレスまたはドメイン名のデフォルトです。
- これは、MetaData:Host = <string> のショートカットです。この入力からのイベントのホストが特定の文字列になるよう設定します。Splunk は、このショートカットが使われたときに自動的に host:: を値の先頭に付け加えます。

Splunk を再起動して、inputs.conf に対して行ったあらゆる変更を有効にします。

別のシステムのデータに対するホストの値を上書きする

中央ログアーカイブで Splunk を実行する、または同一環境の別のホストからコピーされたファイル进行处理する場合、デフォルトの割り当てを上書きする必要があります。その入力に対するすべてのデータのカスタムホスト値または、例えば、異なるサブディレクトリで各ホストに対するログアーカイブを分離するディレクトリ構造を持つ場合など、ソースのパスまたはファイル名が一部一致する部分のいずれかを基にして、入力に対するホスト割り当てを定義できます。

詳しくは、本書の「入力に対するホスト割り当ての設定」を参照してください。

イベントデータを使ってホストの値を上書きする

中央のログホストが Splunk にイベントを送信する場合は、複数のサーバーが関与します。中央のログサーバーは、レポートホストと呼ばれています。イベントが発生したシステムは、元となるホスト(またはホスト)と呼ばれます。この場合、イベント自体の情報を基にホストフィールドの値を設定するルールを定義する必要があります。

詳しくは、本書の「イベントデータを基にしたデフォルトホスト割り当ての上書き」を参照してください。

## 入力に対するホスト割り当ての設定

入力に対するホスト割り当ての設定

特定の条件では、特定の設定入力により Splunk に送られるすべてのデータに対して明示的にホスト値を設定したい場合があります。ホストを静的または動的に設定できます。

- **静的**にホストを設定するとは、指定された入力を通るすべてのイベントに対して同じホストを設定するという事です。
- **動的**にホスト値を設定する場合は、Splunk は、正規表現またはソースの完全ディレクトリパスのセグメントを使って、ソース入力のセグメントからホスト名を抽出します。

同じ入力で異なるソースまたはソースタイプで異なるホストを割り当てるには、本書の「デフォルトホスト割り当ての上書き」を参照してください。

入力のホスト割り当てを静的に設定する

この方法は、入力されるすべてのイベントに対して同じホストを割り当てます。

静的なホスト値の割り当ては、その入力を通る新しいデータにのみ影響を及ぼします。既にインデックスされているデータに対して Splunk Web が表示するホストを訂正する必要がある場合は、ホストにタグを付ける必要があります。

#### Splunk Web の場合

Splunk Web の管理の「データ入力」ページで新しい入力を追加したとき、その入力に対して静的にホストを定義できます。

1. Splunk Web で、画面右上隅の**管理**リンクをクリックします。
2. 管理で、**システムコンフィギュレーションのデータ入力**をクリックします。
3. データ入力ページで、追加または変更する入力タイプを選択します。選択した入力タイプの入力一覧が開きます。
4. ここから、既存の入力を選択して更新する、または**新規**をクリックして選択したタイプで新しい入力を作成します。
5. いずれの方法でも、その入力に対して静的なホスト定義を設定するには、**ホストの設定**ドロップダウンリストから**継続した値**を選択します。
6. **ホストフィールド値**フィールドに入力の静的なホスト値を入力します。
7. 変更を保存します。

入力および入力タイプについては、管理者ガイドの「Splunk の監視事項」を参照してください。

#### 設定ファイルの場合

`inputs.conf` を編集してホスト値を指定します。 `host =` 属性を適切なスタンザに記述します。

`$SPLUNK_HOME/etc/system/local/`、または `$SPLUNK_HOME/etc/apps/` の独自のカスタムアプリケーションディレクトリにある `inputs.conf` を編集します。設定ファイルの全般的な内容については、管理者マニュアルの「設定ファイルについて」を参照してください。

```
[<inputtype>://<path>]
host = $YOUR_HOST
sourcetype = $YOUR_SOURCETYPE
source = $YOUR_SOURCE
```

入力および入力タイプについては、管理者マニュアルの「Splunk の監視事項」を参照してください。

入力に対する静的なホスト割り当ての例

この例では、TCP ポート 9995 の IP アドレス 10.1.1.10 を通過するすべてのイベントを処理します。この入力によるすべてのイベントには、webhead-1 の host 値が割り当てられます。

```
[tcp://10.1.1.10:9995]
host = webhead-1
sourcetype = access_common
source = //10.1.1.10/var/log/apache/access.log
```

入力のホスト割り当てを動的に設定する

この方法は、ソース入力パスのセグメントまたは正規表現のいずれかでホスト名を動的に抽出したい場合に使用します。例えば、インデックスしたい保存ディレクトリがあり、そのディレクトリの各ファイルの名前に関連するホスト情報が含まれている場合は、Splunk を使ってこの情報を抽出して、ホストフィールドに割り当てることができます。

SplunkWeb の場合

前述の Splunk Web による静的なホスト割り当ての設定方法の手順に従ってください。ただし、ホストの設定ドロップダウンリストから継続した値を選択するかわりに、次の 2 つの値のいずれかを選択します。

1. **パスの正規表現** – 正規表現でホスト名を抽出する場合は、このオプションを選択します。正規表現フィールドに抽出するホストに対する正規表現を入力します。
2. **パス上のセグメント** – データソースのパスにあるセグメントからホスト名を抽出する場合は、このオプションを選択します。セグメント #フィールドにセグメントの番号を入力します。例えば、ソースへのパスが /var/log/hostserver で、3 つ目のセグメントをホスト値にする場合は、セグメント #フィールドに 3 を入力します。

設定ファイルの場合

inputs.conf を設定する場合は、動的なホスト抽出を設定できます。SPLUNK\_HOME/etc/system/local/ または、\$SPLUNK\_HOME/etc/apps/ の独自のカスタムアプリケーションディレクトリにある inputs.conf を編集します。設定ファイルの全般的な内容については、管理者マニュアルの「設定ファイルについて」を参照してください。

host\_regex = <regular expression> を追加して、正規表現を使って抽出した値でホストフィールドを上書きします。

```
[<inputtype>://<path>]
host_regex = $YOUR_REGEX
sourcetype = $YOUR_SOURCETYPE
source = $YOUR_SOURCE
```

- 指定がある場合は、正規表現で各入力のファイル名から host 値を抽出します。
- 具体的には、正規表現の最初のグループがホストとして使用されます。
- 正規表現が一致しない場合は、デフォルトの host = 属性がホストに設定されます。

`host_segment = <integer>` を追加して、データソースパスのセグメントを使って抽出された値でホストフィールドを上書きします。

- 指定がある場合は、指定した「/」で分割されたパスのセグメントが各入力のホストとして設定されます。
- 値が整数でない、または 1 より小さい場合は、デフォルトの `host =` 属性がホストに設定されます。

入力に対する動的なホスト割り当ての例

この例では、ファイルパスの正規表現を使用してホストを設定します。

```
[monitor:///var/log]
host_regex = /var/log/(¥w+)
```

この正規表現では、`/var/log/foo.log` からのすべてのイベントが、`foo` の `host` 値となります。

この例では、データソースファイルパスのセグメントを使用してホストを設定します。

```
[monitor://apache/logs/]
host_segment = 3
sourcetype = access_common
```

ここでは、パス `apache/logs` の 3 つ目のセグメントを `host` 値に設定します。

## イベントデータを基にしたデフォルトホスト割り当ての上書き

イベントデータを基にしたデフォルトホスト割り当ての上書き

Splunk は、イベントのデータを基にイベントにデフォルトのホスト名を割り当てます。ここでは、デフォルトの割り当てが正しくない場合に、特定のデフォルトホスト割り当てを上書きする方法について説明します。

デフォルトのホスト割り当てを上書きするには、`transforms.conf` および `props.conf` を編集します。

設定

`transforms.conf` および `props.conf` のソースまたはソースタイプに対して動的に抽出されたホスト名を設定します。  
`$(SPLUNK_HOME)/etc/system/local/` または `$(SPLUNK_HOME)/etc/apps/` の独自のカスタムアプリケーションディレクトリにあるこのファイルを編集します。設定ファイルの全般的な内容については、本書の「設定ファイルについて」を参照してください。

`transforms.conf` の編集

カスタムスタanzaを `$(SPLUNK_HOME)/etc/system/local/transforms.conf` に追加します。スタanzaを以下のように設定します。

```
[$UNIQUE_STANZA_NAME]
DEST_KEY = MetaData:Host
REGEX = $YOUR_REGEX
FORMAT = host::$1
```

スタンザ名および正規表現フィールドに、データに対して正しい値を入力します。

`DEST_KEY = MetaData:Host` を残して `host::` フィールドに値を書き込みます。`FORMAT = host::$1` は、`REGEX` 値を `host::` フィールドに書き込みます。

**注記:** スタンザに固有の識別子となる名前を付けます (`$SPLUNK_HOME/etc/system/default/transforms.conf` のスタンザと間違えないため)

#### props.conf の編集

`$SPLUNK_HOME/etc/system/local/props.conf` でスタンザを作成して、の `props.conf` のソースタイプに対して `transforms.conf` 正規表現を割り当てます。

```
[<spec>]
TRANSFORMS-$name=$UNIQUE_STANZA_NAME
```

<spec> には以下が使えます。

1. <sourcetype>、イベントのソースタイプ。
2. `host::<host>`、<host> はイベントに対するホスト。
3. `source::<source>`、<source> はイベントに対するソース。

`$name` は、変換に使う固有の識別子です。

`$UNIQUE_STANZA_NAME` は、`transforms.conf` で作成した変換のスタンザ名と一致する必要があります。

**注記:** スタンザを定義するとき、任意で、`props.conf` からその他の有効な属性/値ペアを追加します。こうすると、属性を設定した<spec>に割り当てます。例えば、同じ<spec>に設定するカスタム改行ルールがある場合、その属性をスタンザに追加します。

#### 例

`housesness.log` ファイルの次のイベントには、3 つ目にホストが含まれています。

```
41602046:53 accepted fflanda
41602050:29 accepted rhallen
41602052:17 accepted fflanda
```

ホスト値を抽出し、`$SPLUNK_HOME/etc/system/local/transforms.conf` の新しいスタンザに追加するする正規表現を作成します。

```
[housesness]
DEST_KEY = MetaData:Host
REGEX = \s(\w*)$
FORMAT = host::$1
```

ここで、`transforms.conf` スタンザを `$SPLUNK_HOME/etc/system/local/props.conf` とリンクさせて変換を呼び出します。必要に応じて任意で、`props.conf` から追加の属性/値ペアを追加します。

上述の変換は、`props.conf` の以下のスタンザで機能します。

```
[source::.../housesness.log]
TRANSFORMS-rhallen=housesness
SHOULD_LINEMERGE = false
```

上述のスタンザには、追加の属性/値ペア `SHOULD_LINEMERGE = false` があります。これは、Splunk に新しい行に新しいイベントを作成するよう指示します。

**注記：** 属性 `TRANSFORMS-rhallen` にある追加の `-rhallen` は、この変換を別の変換と区別する役割をしています。

この段階でSplunkWebに表示されるイベントは以下のようになります。

```
8 6/22/09 41602052:17 accepted fflanda
4:44:44.000 PM host=fflanda sourcetype=housesness source=./housesness.log
9 6/22/09 41602050:29 accepted rhallen
4:44:44.000 PM host=rhallen sourcetype=housesness source=./housesness.log
10 6/22/09 41602046:53 accepted fflanda
4:44:44.000 PM host=fflanda sourcetype=housesness source=./housesness.log
```

# ソースタイプの取り扱い

## ソースタイプについて

ソースタイプについて

一般的なデータ入力形式は、ソースタイプです。最も代表的なソースタイプは、ログ形式です。例えば、Splunk が自動認識する一般的なソースタイプは以下のとおりです。

- `access_combined`、NCSA 結合型の HTTP ウェブサーバーログ
- `apache_error`、標準の Apache ウェブサーバーエラー
- `cisco_syslog`、PIX ファイアウォール、ルーター、ACS などを含む、Cisco ネットワークデバイスにより生成された標準の syslog、通常リモートの syslog から中央のログホストに送信される
- `websphere_core`、WebSphere から抽出されるコアファイル

**注記：** Splunk が自動認識するソースタイプの詳細一覧は、本書の「ソースタイプの予備練習」を参照してください。

`sourcetype` は、ソースタイプフィールドの名前です。Splunk は、デフォルトで `sourcetype` フィールドを抽出します。つまり、データをインデックスするとき、各イベントに対するソースタイプフィールドを抽出してインデックスします。

`sourcetype` フィールドを使って同様のタイプのデータをあらゆるソースタイプから検索できます。例えば、`sourcetype=weblogic_stdout` を検索して、すべての WebLogic サーバーのイベントを検索します。WebLogic が複数のドメインからログされている場合でも検索します。

ソースとソースタイプ

ソースは、インデックスを持つイベントに対して Splunk が特定するデフォルトフィールドの 1 つです。ソースは、ファイル、ストリーム、特定のイベントが生成するその他の入力の名前です。ファイルおよびディレクトリで監視されるデータの場合、`source` の値は、`/archive/server1/var/log/messages.0` または `/var/log/` などのフルパスです。ネットワークベースのデータソースに対するソースの値は、`UDP:514` などのプロトコルおよびポートです。

**異なるソースから同じソースタイプを持つイベントが作られる場合があります。**例えば、`source=/var/log/messages` を監視し、`udp:514` から直接 `syslog` 入力を受信するとします。`sourcetype=linux_syslog` を検索すると、Splunk はこれらのソース両方からイベントを返します。

Splunk でソースタイプのフィールド値を設定する方法

Splunk は、**ソースタイプ自動認識機能**を使って、受信イベントデータに `sourcetype` 値を設定します。Splunk は、ネットワーク入力のあらゆるファイルまたはストリームの最初の数千行からシグネチャのパターンを計算してインデックス処理中にソースタイプをイベントに割り当てます。このシグネチャは、繰り返し文字パターン、句読点パターン、行の長さなどを特定します。Splunk がシグネチャを計算したら、以前に見られたシグネチャと比較します。シグネチャが根本的に新しいパターンの場合は、Splunk が新しいソースタイプを作成します。`sourcetypes.conf` に新しいパターンの情報を保管します。

ソースタイプ自動認識では期待する結果が得られない場合は、以下を行います。

- **ルールベースのソースタイプ認識を設定して、Splunk が特定するソースタイプの範囲を広げます。**
- **Splunk のソースタイプ自動分類機能を強化して、特定のソースタイプの認識度を高めます。**
- **ソースタイプの自動分類を完全に回避させて、データ入力設定時にソースタイプを設定します。**
- **ソースタイプのタグ付けを使ってインデックスされているソースタイプの名前を変更します。**

ソースタイプの取り扱いに関する詳細は、本書の別のトピックを参照してください。

#### Splunk でソースタイプ値(優先順位)を適用する方法

ユーザーは、Splunk でソースタイプ値をイベントに適用する方法を設定する、または Splunk に自動的に適用させるのいずれかを指定できます。以下のリストは、Splunk でソースタイプ値をイベントに適用する方法とその順序を示しています。

1. `inputs.conf` の入力スタンプ別ソースタイプの詳細仕様：

```
[monitor://$PATH]
sourcetype=$SOURCE
```

2. `props.conf` にスタンプを作成することによる、ソース別のソースタイプの詳細仕様

```
[$SOURCE]
sourcetype=$SOURCE
```

3. ソースタイプのルールベース関連付け：

`props.conf` の `rule::` スタンプに指定した分類ルールを使って、ソースとソースタイプを一致させることができます。

4. 高度な照合：見た目が似ているファイルを照合してソースタイプを作成します。

5. 遅延ルール：

`props.conf` に `[delayedrule::]` スタンプを作成することを除いて、ルールベースの関連性と同様に機能します。これは、Splunk で見逃さないため、「すべてのソースタイプを取り込む」場合に便利です。

6. ソースタイプ自動学習：

Splunk は、ソースタイプが関連付けられていないソースを基に新しいソースタイプを作成します。

## ソースタイプの設定ファイル

ソースのソースタイプは `inputs.conf` に設定します。カスタムインデックスプロパティおよびソースタイプのルールベース関連は `props.conf` を通じて設定します。設定ファイルを手動で変更する前には、必ず設定ファイルについて知っておく必要があります。

## ソースタイプの名前変更

### ソースタイプの名前変更

`props.conf` でソースタイプを設定するとき、ソースタイプの名前を変更できます。複数のソースタイプで同じ名前を共有できます。この方法は、検索するために一連のソースタイプをグループ化する際に便利です。

**注記：** ソースタイプの名前変更は、既にインデックスされたイベントには影響ありません。インデックスされたイベントのソースタイプを変更するには、タグを付けます。詳しくは、本書の「タグとエイリアスについて」を参照してください。

ソースタイプの名前を変更するには、以下をソースタイプスタンザに追加します。

```
[<$SOURCETYPE>]
rename = <string>
```

名前を変更した後は、以下でソースタイプを検索できます。

```
sourcetype=<string>
```

例えば、ソースタイプ `access_combined` を `webaccess` に名前変更する場合は、以下のように記述し、

```
[access_combined]
renamed = webaccess
```

その後、新しいソースタイプ名でイベントを検索するには、以下のように記述します。

```
sourcetype=webaccess
```

**注記：** `props.conf` にソースタイプのインデックスプロパティを設定する場合は、`sourcetypes.conf` に実際に保存されているソースタイプの値を使用する必要があります。

ソースタイプの名前を変更しても、元の名前は削除しません。"`_sourcetype`" 属性を使うと、ソースタイプの元の名前を検索できます。例えば、`access_combined` (ソースタイプの名前を `webaccess` に変更した後)を検索する場合は、以下のように記述します。

```
_sourcetype::access_combined
```

## ルールベースのソースタイプ認識の設定

### ルールベースのソースタイプ認識の設定

ルールベースのソースタイプ認識を設定して、Splunk が認識するソースタイプの範囲を広げます。Splunk は、`props.conf` で指定した正規表現を基にルールベースのソースタイプを自動的に割り当てます。

ソースタイプのルールを設定するには、`$SPLUNK_HOME/etc/system/local/`、または `$SPLUNK_HOME/etc/apps/` の独自のカスタムアプリケーションディレクトリにある `props.conf` を編集します。設定ファイルの全般的な内容については、管理者マニュアルの「設定ファイルについて」を参照してください。

#### 設定

`props.conf` に `rule::` または `delayedrule::` スタンザを追加してルールを作成します。ルールスタンザでは、ソースタイプの名前を宣言します。ソースタイプを宣言した後は、ソースタイプに割り当てるルールを一覧します。ルールは、一連の `MORE_THAN` および `LESS_THAN` 記述を基に作成され、これらは一致する必要があります。記述は、正規表現と一致する指定された行の割合で一致しなければいけない正規表現です。記述はいくつでも指定できます。また、ソースがソースタイプルールに適合するため、すべての記述が一致している必要があります。

以下を `$SPLUNK_HOME/etc/system/local/props.conf` に追加します。

```
[rule::$RULE_NAME] OR [delayedrule::$RULE_NAME]
sourcetype=$SOURCETYPE
MORE_THAN = $REGEX
LESS_THAN = $REGEX
```

**注記：** ルールには、複数の `MORE_THAN` および `LESS_THAN` パターンを持つことができます。ルールが一致するためには、すべてのパターンが適合されている必要があります。

ルールは、指定した文字列を含む行数の割合を基に作成されます。一致するには、ルールがその割合と `MORE_THAN` または `LESS_THAN` のいずれかである必要があります。

#### 例

以下は、`$SPLUNK_HOME/etc/system/default.` の例です。

#### postfix syslog ファイル

```
# postfix_syslog sourcetype rule
[rule::postfix_syslog]
sourcetype = postfix_syslog
# If 80% of lines match this regex, then it must be this type
MORE_THAN_80=^\w{3} +\d+ \d\d:\d\d:\d\d .* postfix(/w+)?\[\d+\]:
```

#### 分割可能テキストの遅延ルール

```
# breaks text on ascii art and blanklines if more than 10% of lines have
# ascii art or blanklines, and less than 10% have timestamps
[delayedrule::breakable_text]
sourcetype = breakable_text
MORE_THAN_10 = (^{?:---|===|\*\*\*|___|=+=))|^\s*$
LESS_THAN_10 = [: ] [012]? [0-9]: [0-5] [0-9]
```

# Splunk のソースタイプ自動分類子の教育

## Splunk のソースタイプ自動分類子の教育

この手順を使って、Splunk で新しいソースタイプを識別するよう教育する、または新しいサンプルを与えて教育済みソースタイプの認識度を高めます。自動分類子の教育を行うと、Splunk で類似するパターンを持つ未来のイベントデータを特定のソースタイプとして分類します。これは、Splunk で混合したソースタイプを持つデータを含むディレクトリ (/var/log など) をインデックスするとき便利です。Splunk は、ほとんどの syslog ファイルに sourcetype=syslog を割り当てる機能で、「教育済み」を実行します。

**注記：** ソースタイプの自動分類子の教育は、将来のイベントデータに適用され、既にインデックスされているイベントデータには適用されませんのでご注意ください。

ハードコードの設定を支持して自動分類子を回避し、入力に対するソースタイプを上書きする、またはソースのソースタイプを上書きするようにできます。または、ルールベースのソースタイプ認識を設定します。

Splunk に内蔵されている匿名ユーティリティを使って、ファイルを匿名にすることもできます。

Splunk が共通形式の認識に失敗する、または不正なソースタイプ値を適用する場合は、その問題を Splunk のサポートに報告し、サンプルファイルを送付してください。

## CLI の場合

ここに、CLI を使ってソースタイプを教育するための入力例を示します。

```
# splunk train sourcetype $FILE_NAME $SOURCETYPE_NAME
```

\$FILE\_NAME にファイルまでの全パスを入力します。\$SOURCETYPE\_NAME は、ユーザーが作成するカスタムソースタイプです。

一般的に、新しいソースタイプに対して小数の異なるサンプルを使って教育し、Splunk がソースタイプの違いを学べるようにすることが大切です。

## 教育済みソースタイプ

### 教育済みソースタイプ

Splunk は、教育済みのソースタイプを送って多くの異なるソースタイプを識別します。ソースタイプの数は、自動的かつ適切に認識、タグ付け、および構文解析されます。また、自動認識されないが SplunkWeb または inputs.conf で割り当て可能な大量の教育済みソースタイプを保持しています。

Splunk が教育済みソースタイプに対して最適化されたインデックスプロパティを持つため、データと一致する場合は、教育済みのソースタイプを使うと便利です。ただし、データがどの教育済みソースタイプにも適合しない場合は、カスタムプロパティを持たないデータの形式を仮想インデックスすることができます。

ソースタイプおよびその仕組みについて詳しくお読みください。

自動認識されたソースタイプ

ソースタイプ名	起源	例
access_combined	NCSA 結合型形式 http ウェブサーバー ログ(アパッチまたは その他のウェブサー バで生成可能)	10.1.1.43 - webdev [08/Aug/2005:13:18:16 "-" "check_http/1.10 (nagios-plugins 1.4)"
access_combined_wcookie	NCSA 結合型形式 http ウェブサーバー ログ(アパッチまたは その他のウェブサー バで生成可能)、末尾 に cookie フィールド を付加	"66.249.66.102.1124471045570513" 59.92.110.121 -0700] "GET /themes/splunk_com/images/logo_"http://www.splu nk.org/index.php/docs" "en-US; rv:1.7.8) Gecko/20050524 Fedora/1.0.4-"61.3.110.148.1124404439914689"
access_common	NCSA 共有型形式 http ウェブサーバー ログ(アパッチまたは その他のウェブサー バで生成可能)	10.1.1.140 - - [16/May/2005:15:01:52 -0700] /themes/ComBeta/images/bullet.png HTTP/1.1"
apache_error	標準 Apache ウェブサ ーバーエラーログ	[Sun Aug 7 12:17:35 2005] [error] [client /home/reba/public_html/images/bullet_image
asterisk_cdr	標準アスタリスク IP PBX 呼び出し詳細レ コード	"" , "5106435249" , "1234" , "default" , "" "Jam es Jesse"" <5106435249> , "SIP/5249-1ce3" , "" , "15:19: 25" , "2005-05-26 15:19:25" , "2005-05-15:19:42" , 17 , 17 , "ANSWERED" , " DOCUMENTATION"
asterisk_event	標準アスタリスクイ ベントログ(管理イベ ント)	Aug 24 14:08:05 asterisk[14287]: Manager
asterisk_messages	標準アスタリスクメ ッセージログ(エラー と警告)	Aug 24 14:48:27 WARNING[14287]: Channel 'Zap/1-1' sent into invalid extension 's' in context 'default', but no invalid handler

asterisk_queue	標準アスタリスクキューログ	NONE   NONE   NONE   CONFIGRELOAD
cisco_syslog	ルータ、ACS などを含む Cisco ネットワークデバイスにより生成された標準 Cisco Syslog 通常、リモート syslog から中央ログホストに送信	Sep 14 10:51:11 stage-test.splunk.com Aug Inbound TCP connection denied from IP_addr/TCP_flags on interface int_name Inbound 144.1.10.222/9876 to 10.0.253.252/6161 flags
db2_diag	標準 IBM DB2 データベースの管理およびエラーログ	2005-07-01-14.08.15.304000-420 I27231H328 4760 PROC : db2fmp.exe INSTANCE: DB2 NODE Table Maintenance, db2HmonEvalStats, probe:evaluation has finished on database TRADEDB
exim_main	Exim MTAのメインログ	2005-08-19 09:02:43 1E69KN-0001u6-8E => R=send_to_relay T=remote_smtp H=mail.int.
exim_reject	Exim の拒否ログ	2005-08-08 12:24:57 SMTP protocol violation: sent without waiting for greeting): rejected H=gate.int.splunk.com [10.2.1.254]
linux_messages_syslog	標準 linux syslog (ほとんどのプラットフォームの /var/log/messages)	Aug 19 10:04:28 db1 sshd(pam_unix) [15979]: session opened for user root by (uid=0)
linux_secure	Linux securelog	Aug 18 16:19:27 db1 sshd[29330]: Accepted publickey for root from ::ffff:10.2.1.5 port 40892 ssh2
log4j	log4j を使った J2EE サーバー生成の Log4j 標準出力	2005-03-07 16:44:03,110 53223013 [PoolThread-0] INFO [STDOUT] got some property...
mysqld_error	標準 mysql エラーログ	050818 16:19:29 InnoDB: Started; log sequence number 0 43644 /usr/libexec/mysqld: ready for connections. Version: '4.1.10a-log' socket: '/var/lib/mysql/mysql.sock' port: 3306 Source distribution
mysqld	標準 mysql クエリログ	53 Query SELECT xar_dd_itemid, xar_dd_propid, xar_dd_value FROM xar_dynamic_data WHERE

	グ、テキストへの変換後の mysql のバイナリログと一致	xar_dd_propid IN (27) AND xar_dd_itemid = 2
postfix_syslog	Unix/Linux syslog 工場のレポートによる標準 Postfix MTA ログ	Mar 1 00:01:43 avas postfix/smtpd[1822]: 0141A61A83: client=host76-117.pool80180.interbusiness.it[80.180.117.76]
sendmail_syslog	Unix/Linux syslog 工場のレポートによる標準 Sendmail MTA ログ	Aug 6 04:03:32 nmrjl00 sendmail[5200]: q64F01Vr001110: to=root, ctladdr=root (0/0), delay=00:00:01, xdelay=00:00:00, mailer=relay, min=00026, relay=[101.0.0.1] [101.0.0.1], dsn=2.0.0, stat=Sent (v00F3HmX004301 Message accepted for delivery)
sugarcrm_log4php	log4php ユーティリティを使用したレポートによる標準 Sugarcrm アクティビティログ	Fri Aug 5 12:39:55 2005,244 [28666] FATAL layout_utils - Unable to load the application list language file for the selected language(en_us) or the default language(en_us)
weblogic_stdout	標準ネイティブ BEA フォーマットの Weblogic サーバログ	####<Sep 26, 2005 7:27:24 PM MDT> <Warning> <WebLogicServer> <bea03> <asiAdminServer> <ListenThread.Default> <<WLS Kernel>> <> <BEA-000372> <HostName: 0.0.0.0, maps to multiple IP addresses:169.254.25.129,169.254.193.219>
websphere_activity	Websphere アクティビティログ、サービスログとして参照	ComponentId: Application Server ProcessId: 2580 ThreadId: 0000001c ThreadName: Non-deferrable Alarm : 3 SourceId: com.ibm.ws.channel.framework.impl.WSChannelFrameworkImpl ClassName: MethodName: Manufacturer: IBM Product: WebSphere Version: Platform 6.0 [BASE 6.0.1.0 o0510.18] ServerName: nd6Cell101\was1Node01\TradeServer1 TimeStamp: 2005-07-01 13:04:55.187000000 UnitOfWork: Severity: 3 Category: AUDIT PrimaryMessage: CHF0020I: The Transport Channel Service has stopped the Chain labeled SOAPAcceptorChain2 ExtendedMessage:
websphere_core	Websphere の Corefile エクスポート	NULL----- -----0SECTION TITLE subcomponent dump routine NULL===== 1TISIGINFO signal 0 received 1TIDATETIME Date: 2005/08/02 at 10:19:24 1TIFILENAME Javacore filename: /kmbcc/javacore95014.1122945564.txt NULL 0SECTION XHPI subcomponent dump routine NULL

		===== 1XHTIME Tue Aug 2 10:19:24 20051XHSIGRECV SIGNONE received at 0x0 in <unknown>. Processing terminated. 1XHFULLVERSION J2RE 1.3.1 IBM AIX build ca131-20031105 NULL
<b>websphere_trlog_syserr</b>	<b>IBM のネイティブ tr ログ形式の標準 Websphere システム エラーログ</b>	[7/1/05 13:41:00:516 PDT] 000003ae SystemErr R at com.ibm.ws.http.channel. inbound.impl.HttpICLReadCallback.complete (HttpICLReadCallback.java(Compiled Code)) (truncated)
<b>websphere_trlog_sysout</b>	<b>IBM のネイティブ tr ログ標準 Websphere システム出力ログ、 Resin および Jboss に 対する log4j サーバ ーログと同様、システ ムエラーログとして のサンプルフォーマ ット(重大度、情報性 の低いイベント)</b>	[7/1/05 13:44:28:172 PDT] 0000082d SystemOut O Fri Jul 01 13:44:28 PDT 2005 TradeStreamerMDB: 100 Trade stock prices updated: Current Statistics Total update Quote Price message count = 4400 Time to receive stock update alerts messages (in seconds): min: -0.013 max: 527.347 avg: 1.0365270454545454 The current price update is: Update Stock price for s:393 old price = 15.47 new price = 21.50
<b>windows_snare_syslog</b>	<b>第三機関 Intersect Alliance Snare エー ジェントにより Unix または Linuxserver の リモート syslog にレポートされた標 準 Windows イベント ログ</b>	0050818050818 Sep 14 10:49:46 stage-test.splunk.com Windows_Host MSWinEventLog 0 Security 3030 Day Aug 24 00:16:29 2005 560 Security admin4  User Success Audit Test_Host Object Open: Object Server: Security Object  Type: File Object Name: C:\Directory\secrets1.doc New Handle ID: 1220  Operation ID: {0,117792} Process ID: 924 Primary User Name: admin4 Primary  Domain: FLAME Primary Logon ID: (0x0,0x8F9F) Client User Name: - Client  Domain: - Client Logon ID: - Accesses SYNCHRONIZE ReadData (or ListDirectory) Privileges -Sep

## 教育済みソースタイプ

このリストには、自動認識されるソースタイプと自動認識されない教育済みソースタイプの両方が記載されています。

カテゴリ	ソースタイプ
アプリケーションサーバー	log4j, log4php, weblogic_stdout, websphere_activity, websphere_core, websphere_trlog
データベース	mysqld, mysqld_error, mysqld_bin
電子メール	exim_main, exim_reject, postfix_syslog, sendmail_syslog, procmail
オペレーティングシステム	linux_messages_syslog, linux_secure, linux_audit, linux_bootlog, anaconda, anaconda_syslog, osx_asl, osx_crashreporter, osx_crash_log, osx_install, osx_secure, osx_daily, osx_weekly, osx_monthly, osx_window_server, windows_snare_syslog, dmesg, ftp, ssl_error, syslog, sar, rpmpkgs
ネットワーク	novell_groupwise, tcp
プリンタ	cups_access, cups_error, spooler
ルーターとファイアウォール	cisco_cdr, cisco_syslog, clavister
VoIP	asterisk_cdr, asterisk_event, asterisk_messages, asterisk_queue
ウェブサーバー	access_combined, access_combined_wcookie, access_common, apache_error, iis
その他	snort

## ソースタイプ自動割当の回避

### ソースタイプ自動割当の回避

入力設定時にソースタイプを設定して特定のデータ入力に対するソースタイプ自動割り当てを上書きできます。(下参照) ただし、この方法は、精度が高くないため、同じホストまたはソースからのデータにすべて同じソースタイプ名が割り当てられます。

1つのディレクトリ入力で異なるソース名を与える必要がある場合は、1つのソースに対するソースタイプを設定します。

## 入力に対するソースタイプの上書き

この手順を使って、入力によるすべてのデータのソースタイプを明確に設定します。

ディレクトリ (/var/log/ など) を入力する場合は、この方法でそのディレクトリ内のすべてのファイルに対して同じソースタイプを割り当てます。同じ入力ディレクトリ内にある個々のソースに異なるソースタイプを割り当てるには、ソースに対してソースタイプを設定します。

**注記：** この設定は、新しい受信データにのみ影響を及ぼします。Splunk Web で表示される既にインデックスされているデータのソースタイプを修正するには、そのソースタイプにタグを作成します。

## Splunk Web の場合

Splunk Web でデータ入力を設定するときに、ソースタイプをハードコード化できます。

### ソースタイプリストから選ぶ

ソースが Splunk の教育済みソースタイプの 1 つである場合は、同じ名前を選択して Splunk に自動割り当てさせる方法が適しています。Splunk の教育済みソースタイプの説明は、教育済みソースファイルのリファレンスリストを参照してください。

ソースタイプ設定のドロップダウンからリストからを選択します。

### 新しいソースタイプ名を使う

データ入力画面下部のドロップダウンメニューから **マニュアル** を選択します。

ソースタイプボックスにソースタイプ名を入力します。

ここで、イベントに `sourcetype=` 値が追加されます。

## 設定ファイルの場合

`inputs.conf` で入力を設定するときに、`sourcetype` を設定することもできます。 `sourcetype =` 属性を `$(SPLUNK_HOME)/etc/system/local/inputs.conf` の適切なスタンザに含めます。

```
[tcp://:9995]
  connection_host = dns
  sourcetype = log4j
  source = tcp:9995
```

ここで、ポート 9995 の TCP 入力を通してイベントに `sourcetype=log4j` を設定します。

## ソースのソースタイプを上書き

この手順を使って、`props.conf` のソースを基にソースタイプを割り当てます。 `$(SPLUNK_HOME)/etc/system/local/` または `$(SPLUNK_HOME)/etc/apps/` の独自のカスタムアプリケーションディレクトリにある `props.conf` ファイルを編集します。設定ファイルの全般的な内容については、設定ファイルのしくみを参照してください。

**注記：** これは、設定変更した後に入力される新しいデータにのみ影響します。Splunk Web に表示される既にインデックスされたデータのソースタイプを修正したい場合は、ソースタイプにタグを作成します。

設定ファイルの場合

`$SPLUNK_HOME/etc/system/local/props.conf` にソースのスタンザを追加して、`sourcetype = 属性` を設定します。

```
[source::.../var/log/anaconda.log(.\d+)?]
sourcetype = anaconda
```

ここで、文字列 `/var/log/anaconda.log` の後に数字文字を含むソースのイベントを `sourcetype=anaconda` に設定します。

Splunk では、スタンザのソースパスの正規表現 (`[source::.../web/...log]` など) は、できる限り具体的に表記し、絶対に正規表現が `"..."` で終わらないよう推奨しています。例えば、以下は悪い例です。

```
[source::/home/fflanda/...]
sourcetype = mytype
```

この例では、`/home/fflanda` の `gzip` ファイルは `gzip` ファイルではなく `mytype` ファイルとして処理されるため、危険です。

この場合は、以下のように記述します。

```
[source::/home/fflanda/...log(.\d+)?]
sourcetype = mytype
```

`props.conf` について詳しくお読みください。

## props.conf でソースタイプ設定を指定

`props.conf` でソースタイプ設定を指定

`props.conf` ではソースタイプの詳細設定ができます。以下の属性/値ペアを使ってソースタイプの設定を指定します。ソースタイプスタンザを `$SPLUNK_HOME/etc/system/local/`、または `$SPLUNK_HOME/etc/apps/` の独自のカスタムアプリケーションディレクトリにある `props.conf` ファイルに追加します。設定ファイルについては、設定ファイルのしくみを参照してください。

**注記：** 以下の属性/値ペアは、`[<$SOURCETYPE>]` で始まるスタンザにのみ設定します。

```
invalid_cause = <string>
```

- `[<sourcetype>]` スタンザにのみ設定可能です。
- Splunk は `invalid_cause` セットではデータをインデックスしません。
- `<string>` を `"archive"` に設定して、ファイルをアーカイブプロセッサ(`unarchive_cmd` で指定)に送信します。
- Splunklogger をデバッグモードで実行している場合は、`splunkd.log` にエラーを投じるようその他の文字列へ設定します。
- デフォルトは空白です。

unarchive\_cmd = <string>

- invalid\_cause を"archive"に設定した場合にのみ呼び出されます。
- <string> は、シェルコマンドを指定して、アーカイブソースの抽出を実行します。
- 必ず stdin の入力を行い、stdout の出力を生成するシェルコマンドを実行します。
- バッチ処理ファイルは使用しないでください。 preprocessing\_script を使用します。
- デフォルトは空白です。

LEARN\_MODEL = <true/false>

- 周知のソースタイプの場合は、fileclassifier がモデルファイルを学習ディレクトリに追加します。
- 各種ソースタイプ(ソースタイプの作成の良い例ではないソースコードなど)に対する動作を無効にする場合は、LEARN\_MODEL = false を設定します。
  - ◆ ◆具体的には、ソースを名前やルールなどで簡単に分類でき、コンテンツを分析しても得るものがない場合は、LEARN\_MODEL を false に設定します。
- デフォルトは空白です。

maxDist = <integer>

- ソースタイプモデルが現在のファイルと異なる度合いを決めます。
- 値が大きいくほど、許容範囲が広がります。
- 例えば、値が小さい場合(10 など)は、指定したソースタイプの違いも少なくなります。
- 大きい値は、特定のソースタイプのファイルが大幅に異なることを示します。
- デフォルトは 300 です。

# イベントタイプの管理

## イベントタイプについて

### イベントタイプについて

イベントタイプは、データを理解しやすくするための分類システムです。イベントタイプを使うと、大量のデータの処理、類似パターンの検索、アラートやレポートの作成などが行えます。

### イベントとイベントタイプ

イベントは、ログファイルに記載される活動を示す 1 つのレコードです。一般的にイベントには、タイムスタンプが記載され、監視またはログ記録されているシステムの状態に関する情報を提供します。

イベントタイプは、イベントをカテゴリー分類することにより検索を簡素化するためにユーザーが定義するフィールドです。イベントタイプを使うと、共通の特性を持つイベントを分類することができます。検索結果が返ると、周知のイベントタイプと照合チェックされます。イベントタイプは、eventtypes.conf のイベントタイプ定義と一致するイベントがある場合に、検索時間にイベントに適用されます。データをインデックスしてから、イベントタイプにタグを付ける、または保存します。

### イベントタイプの分類

独自のイベントタイプを作成する方法はいくつかあります。Splunk Web または設定ファイルを使ってイベントタイプを定義する、または検索をイベントタイプとして保存することもできます。検索をイベントタイプとして保存する場合は、punct フィールドを使って検索を作成できます。punct フィールドは、イベントの構造を基に検索の絞込みを手助けします。

### punct フィールドを使った類似イベントの検索

イベントの形式はイベントタイプに固有のため、Splunk では、イベントの句読点文字を punct と呼ばれるフィールドにインデックスします。punct フィールドは、イベントの最初の行から 30 の句読点文字を保存します。このフィールドは、同類のイベントを素早く検索する場合に役立ちます。

### punct の使用に関する注意事項

- 引用符およびバックスラッシュは無視されます。
- スペースは、アンダーライン(\_)に置き換えられます。
- タブは "t" に置き換えられます。
- アルファベット文字に続くダッシュは無視されます。
- 対象となる句読点文字：  
",;~#\$%&+. /: = ? @ \ \ ' | \* \ n \ r \ " ( ) { } < > [ ] ^ ! "
- punct フィールドは、生成時に PKI を使って署名されている、\_audit インデックスのイベントには使えません。

punct フィールドの使い方およびその他のイベント識別方法については、ユーザーマニュアルの「類似するイベントを分類してグループ化する」を参照してください。

#### Punct の例

以下のイベントでは、

```
####<Jun 3, 2005 5:38:22 PM MDT> <Notice> <WebLogicServer> <bea03> <asiAdminServer>  
<WrapperStartStopAppMain> <>WLS Kernel<> <> <BEA-000360> <Server started in RUNNING mode>
```

以下の句読点が生成されます。

```
####<_, __: __>_<>_<>_<>_<>_
```

以下のイベントでは、

```
172.26.34.223 - - [01/Jul/2005:12:05:27 -0700] "GET  
/trade/app?action=logout HTTP/1.1" 200 2953
```

以下の句読点が生成されます。

```
..._--_[::_-]_\"_?=_/.\"_
```

#### イベントタイプのディスカバリ

任意の検索を `typelearner` コマンドへパイプして、Splunk Web で直接イベントタイプを作成します。

`eventdiscoverer.conf` ファイルは、ほとんどの場合使用されませんが、Splunk Web で新しいイベントタイプを学習するときに無視する用語を指定することができます。

#### 新しいイベントタイプの作成

最も簡単に新しいイベントタイプを作成するには、Splunk Web を使います。検索を保存するのと同じ方法でイベントタイプを保存します。イベントタイプの保存について詳しくお読みください。

`eventtypes.conf` を変更して新しいイベントタイプを作成します。検索をイベントタイプとして保存する方法については、ユーザーマニュアルの「類似するイベントを分類してグループ化する」を参照してください。

#### イベントタイプのタグ

イベントタイプにタグを付けてデータをカテゴリー分類します。1つのイベントに複数のタグを付けることができます。イベントタイプへのタグ付けについては、本書の「イベントタイプのタグ付け」を参照してください。

イベントタイプの設定ファイル

イベントタイプは `eventtypes.conf` に保存されます。

イベントタイプディスカバリの用語は、`eventdiscoverer.conf` に設定されます。

## Splunk Web によるイベントタイプの定義

Splunk Web によるイベントタイプの定義

ほとんどの検索はイベントタイプとして保存できます。1つのイベントが複数のイベントタイプを持つこともできます。Splunk Web で作成したイベントタイプは、`$SPLUNK_HOME/etc/system/local` または `$SPLUNK_HOME/etc/apps/` にある独自のアプリケーションディレクトリの `eventtypes.conf` に自動追加されます。(カスタマイズしたデータを別のサーバーに簡単に転送したい場合は、後者を使用してください。)

**注記：** インデックス、`hosttag`、`eventtypetag`、`sourcetype`、またはパイプ演算子を指定して検索するイベントタイプは作成できません。

検索をイベントとして保存

検索をイベントとして保存するには以下を行います。

- 検索を実行します。
- **アクション...** ドロップダウンを選択して、**イベントタイプとして保存...** をクリックします。

検索用語が予め入力された **イベントタイプを保存** ダイアログボックスが現れます。

- イベントタイプに名前を付けます。
- 任意で、イベントタイプのタグをコンマ区切りで1つまたは複数追加します。
- **保存** をクリックします。

ここから、イベントタイプを検索で使用できるようになります。

```
eventtype=foo
```

## eventtypes.conf に直接イベントタイプを設定

eventtypes.conf に直接イベントタイプを設定

`eventtypes.conf` を設定して新しいイベントタイプを追加、または既存のイベントタイプを更新できます。いくつかのデフォルトのイベントタイプは、`$SPLUNK_HOME/etc/system/default/eventtypes.conf` に定義されています。Splunk Web で作成したイベントタイプは、`$SPLUNK_HOME/etc/system/local/eventtypes.conf` に自動追加されます。

## 設定

eventtypes.conf のイベントタイプに変更を加えます。例えば、

`$$SPLUNK_HOME/etc/system/README/eventtypes.conf.example` を使う、または自分専用の `eventtypes.conf` を作成します。

`$$SPLUNK_HOME/etc/system/local/`、または `$$SPLUNK_HOME/etc/apps/` の独自のカスタムアプリケーションディレクトリにある `eventtypes.conf` を編集します。設定ファイルの全般的な内容については、管理者マニュアルの「設定ファイルについて」を参照してください。

[`$$EVENTTYPE`]

- イベントタイプのヘッダーです。
- `$$EVENTTYPE` は、イベントタイプの名前です。
  - ◆ ◆ イベントタイプはいくつでも持つことができます。それぞれがスタンザおよび複数の以下の属性/値ペアで表されます。
- **注記:** イベントタイプの名前にパーセント文字で囲まれたフィールド名がある場合 (`$$FIELD%` など)、`$$FIELD` の値は、検索時間でそのイベントのイベントタイプ名と置換されます。例えば、イベントタイプのヘッダー `[cisco-code]` に `code=432` がある場合は、`</code>[cisco-432]</code>` に置換されます。

`search = <string>`

- このイベントタイプの検索条件です。
- 例: `error OR warn`
- **注記:** インデックス、`hosttag`、`eventtypetag`、`sourcetype`、またはパイプ演算子を指定して検索するイベントタイプは作成できません。

`tags = <string>`

- イベントタイプにタグを付ける際に使われるスペース区切りの単語

`isglobal = <1 or 0>`

- イベントタイプの共有を切り替えます。
- `isglobal` が 1 に設定されている場合は、誰でもこのイベントを見るまたは使うことができます。
- デフォルトは 1 です。

`disabled = <1 or 0>`

- イベントタイプのオン/オフを切り替えます。
- 1 と設定して無効にします。

## 例

ここに、`web` と `fatal` と呼ばれる 2 つのイベントタイプがあります。

[`web`]

`search = html OR http OR https OR css OR htm OR html OR shtml OR xls OR cgi`

```
[fatal]
search = FATAL
```

#### イベントタイプの無効化

`disabled = 1` をイベントタイプスタンプ `eventtypes.conf` に追加してイベントタイプを無効にします。

```
[$EVENTTYPE]
disabled = 1
```

`$EVENTTYPE` は、無効にするイベントタイプの名前です。

`web` イベントタイプを無効にする場合は、次のように記述します。

```
[web]
disabled = 1
```

## イベントタイプテンプレートの設定

#### イベントタイプテンプレートの設定

イベントタイプテンプレートは、検索時間のイベントタイプを作成します。 `eventtypes.conf` にイベントタイプテンプレートを定義します。 `$SPLUNK_HOME/etc/system/local/`、または `$SPLUNK_HOME/etc/apps/` の独自のカスタムアプリケーションディレクトリにある `eventtypes.conf` を編集します。

設定ファイルの全般的な内容については、管理者マニュアルの「設定ファイルについて」を参照してください。

#### イベントタイプテンプレートの設定

イベントタイプテンプレートは、パーセント文字で囲まれたフィールド名を使って、`;%FIELD%` 値をイベントタイプの名前と置換する検索時間のイベントタイプを作成します。

```
[$NAME-;%FIELD%]
$SEARCH_QUERY
```

つまり、テンプレートの検索クエリが `;%FIELD%=bar` のイベントを返す場合は、Splunk がそのイベントに対して、`$NAME-bar` というタイトルのイベントタイプを作成します。

#### 例

```
[cisco-;%code%]
search = cisco
```

"cisco" の検索で `code=432` を持つイベントが返されると、Splunk は、タイトルを "cisco-432" にしたイベントタイプを作成します。

# タグとエイリアスの定義

## タグとエイリアスについて

### タグとエイリアスについて

データには、関連したフィールド値を持つイベントのグループがある場合があります。このように特定のイベントデータのグループを効率よく検索する手助けとして、フィールド値にタグを割り当てることができます。さまざまな抽出フィールド(イベントタイプ、ホスト、ソース、ソースタイプなど)に複数のタグを割り当てることができます。

タグは以下の場合に使用できます。

- **要約フィールド値**(IP アドレス、ID 番号など)の追跡を手助けします。例えば、本社に関連する IP アドレスの値を 192.168.1.2 とします。その IPaddress 値に *mainoffice* というタグを付けると、そのタグを検索してその IP アドレスを持つイベントを見つけます。
- **1 つのタグを使用して一連のフィールド値をグループにまとめると**、1 つのコマンドでそれらを検索できます。例えば、2 つのホスト名が同じコンピュータに関連付けられているとします。この値に同じタグを付けることができます。そのタグを検索すると、Splunk が両方のホスト名が関わるイベントを返します。
- **条件が異なる複数のタグを具体的な抽出フィールドに与えると**、タグベースの検索を実行して、期待する結果を素早く得ることができます。この仕組みを理解するには、以下の例を参照してください。

例：

企業イントラネット内でデータソースの IP アドレスを参照する IPaddress と呼ばれる抽出フィールドがあります。機能または場所を基に各 IP アドレスにタグをつけると、この IPaddress を便利に活用できるようになります。すべてのルーターの IP アドレスに *router* というタグを付けたり、設置場所を基に IP アドレスに、例えば *SF* や *Building1* などのタグを付けたりできます。サンフランシスコの *Building 1* に設置されているルーターの IP アドレスに、*router*、*SF*、*Building1* のタグが付けられます。

サンフランシスコで *Building1* 以外に設置されているすべてのルーターを検索するには、以下のように記述します。

```
tag=router tag=SF NOT (tag=Building1)
```

## フィールドのエイリアス作成

### フィールドのエイリアス作成

1 つのフィールドに複数のエイリアスが作成できます。元のフィールドは削除されません。この処理を行うと、エイリアスを使って元のフィールドを検索できます。

**重要：** フィールドエイリアスは、キー/値の抽出後、フィールド検索の前に行われます。したがって、フィールドエイリアスを基にした検索テーブルの指定が可能です。これは、検索テーブルにデータのフィールドと同じフィールドが複数あり、それぞれが別の名前を持つ場合に便利です。詳しくは、本書の「外部データソースのフィールド検索」を参照してください。

エイリアスは、インデックスタイムおよび検索時間の双方で抽出されたフィールドに定義できます。

`$SPLUNK_HOME/etc/system/local/`、または `$SPLUNK_HOME/etc/apps/` の独自のカスタムアプリケーションディレクトリで編集する `props.conf` にフィールドエイリアスを追加します。(カスタマイズしたデータを別のインデックスサーバーに簡単に転送したい場合は、後者を使用してください。)

フィールドエイリアスは以下の手順で行います。

1. `props.conf` のスタンザに以下の行を追加します。

```
FIELDALIAS-<class> = (<orig_field_name> AS <new_field_name>)+
```

- `<orig_field_name>` は、フィールドの元の名前です。
- `<new_field_name>` は、フィールドに割り当てられるエイリアスです。
- 1つのスタンザに複数のフィールドエイリアスを含めることもできます。

2. Splunk を再起動して変更を有効にします。

検索に追加するフィールドエイリアスの例

"ip" を "ipaddress" として参照して検索時間に抽出したフィールドの外部固定テーブル CSV ファイルの検索を作成しているとします。抽出を定義した `props.conf` ファイルに、"ipaddress" を "ip" のエイリアスとする行を以下のように追加します。

```
[accesslog]
EXTRACT-extract_ip = (?<ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})
FIELDALIAS-extract_ip = ip AS ipaddress
```

`props.conf` で検索を設定するとき、`ip` を使う代わりに `ipaddress` を使用します。

```
[dns]
lookup_ip = dnsLookup host OUTPUT ipaddress
```

検索時間のフィールド抽出については、本書の「検索時間でフィールド追加」を参照してください。

フィールド検索については、本書の「外部データソースのフィールド検索作成」を参照してください。

## ホストフィールドのタグ付け

ホストフィールドのタグ付け

ホストフィールドにタグを付けると、ナレッジキャプチャ、共有、およびより正確な検索の作成などに役立ちます。ホストフィールドは、複数の単語でタグ付けが可能です。この機能を使って、機能または種類でホストをグループ化したり、同類のサーバーグループのすべてのアクティビティを簡単に検索したりできます。特定の入力のホストフィールドの値が変っている場合は、新しいホスト名で既にインデックスされているイベントにタグを付けて、データセットの検索を簡素化できます。

Splunk Web でホストフィールドにタグを追加

Splunk Web でホストフィールドにタグを追加するには、以下の操作を行います。

1. タグを付けるホストでデータ検索を実行します。
2. ホストフィールド横のドロップダウン矢印を使って **Tag host=<current host value>** を選択します。
3. コンマ区切りでタグを入力します。

ホスト名とタグ付きホストフィールド

ホストフィールドの値は、イベントをインデックスするときに設定されます。この値は、Splunk サーバーのホスト名を基にデフォルト設定される、入力して設定する、または各イベントデータから抽出されます。別のホスト名でホストフィールドにタグを付けてもホストフィールドの実値は変わりません。検索時は、ホストフィールドの値ではなく、指定したタグを使用します。各イベントは 1 つしかホスト名を持つことはできませんが、ホストタグは複数持つことができます。

例えば、Splunk サーバーが特定のホストからコンプライアンスデータを受信する場合、そのホストに **compliance** タグを付けると、コンプライアンスの検索が簡単になります。ホストタグを使うと、基本となるホスト名をマスキングしたり、変更したりする必要なく、自由にデータグループが作成できます。

特定の入力ソースのデータをインデックスした後、その入力のホストフィールドの値を変更する場合、ホストフィールドに別のホスト名でタグ付けすると、その入力による新しいデータすべてが、新しいホストフィールド値を持つことができ、インデックスに既存のデータは古い値を維持します。既存のデータのホストフィールドにタグを付けると、既存のデータすべてを除外することなく、新しいホスト値を検索することができます。

## イベントタイプのタグ

イベントタイプのタグ

イベントタイプにタグを付けて、データに情報を追加します。すべてのイベントタイプが複数のタグを持つことができます。例えば、すべてのファイアウォールイベントタイプに **firewall** のタグを付け、ファイアウォールイベントタイプのサブセットに **deny** および別のサブセットに **allow** のタグを付けることができます。イベントタイプにタグが付けられると、タグ付けされたパターンに一致するすべてのイベントタイプにタグが付けられます。

**注記：** Splunk Web でイベントを作成または `eventtypes.conf` でイベントを設定したときにタグを付けることができます。

管理を使ったイベントタイプへのタグの追加

Splunk 管理では、イベントタイプの一覧表示と編集ができます。

- 右上隅の**管理**リンクをクリックします。

- イベントタイプを選択します。
- タグを付けるイベントタイプを探し、名前をクリックして詳細ページに移動します。
  - ◆ 注記: イベントタイプには特定の Splunk アプリケーションに関連付けられている場合があるので注意が必要です。役割ベースの権限により、イベントタイプの表示および編集が制限されている場合があります。
- イベントタイプの詳細ページで、タグフィールドにタグを追加または編集します。
- 保存をクリックして変更を確認します。

イベントタイプにタグを付けた後は、`tag::=<tagname` または `tag=<tagname>` の構文を検索バーに入力して検索することができます。

```
tag=foo tag::host=*local*
```

# イベントをトランザクションにグループ化

## トランザクションについて

### トランザクションについて

トランザクションは、時間を計る概念的に関連したイベントのグループです。トランザクションタイプは、設定されたトランザクションで、Splunk にフィールドとして保存されます。複数のデータソースが複数のログエントリによりトランザクションを生成します。

例えば、顧客がオンラインストアで買い物をすると、複数のソースに渡ってトランザクションが生成されます。ウェブアクセスイベントは、アプリケーションサーバーログのイベントと、セッション ID を共有する場合があります。アプリケーションサーバーログには、アカウント ID、トランザクション ID、製品 ID などが含まれ、トランザクション ID は、メッセージ ID のメッセージキューに存在し、現実のアプリケーションは、配送状況と共にメッセージ ID をログしている場合があります。このようなすべてのデータが 1 つのユーザートランザクションを表しています。

以下の例は、トランザクションの一部です。

- ウェブアクセスイベント
- アプリケーションサーバーイベント
- ビジネストランザクション
- 電子メール
- セキュリティ違反
- システム障害

### トランザクション検索

トランザクション検索は、複数のイベントログにまたがる物理的なイベントを一覧するという意味で便利です。トランザクションコマンドを使用して、トランザクションを定義する、または `transactiontypes.conf` に指定されているトランザクションオプションを上書きします。

詳しくは、本書の「トランザクションの検索」を参照してください。

### トランザクションタイプの設定

作成したトランザクション検索を保持したい場合があります。または、持続的なトランザクションタイプを作成したい場合があります。`transactiontypes.conf` を編集してトランザクションを保存できます。スタンプを作成し、仕様を一覧してトランザクションを定義します。

トランザクションタイプの設定については、本書の「トランザクションの定義」をお読みください。

## トランザクションの検索

### トランザクションの検索

Splunk Web、または CLI のトランザクション検索コマンドを使ってトランザクションを検索します。transaction コマンドは、レポートに使用可能なイベントのグループを作成します。transaction を使用するには、トランザクションタイプ (transactiontypes.conf で設定) を呼び出す、または transaction コマンドの検索オプションを設定して検索にトランザクション制約を定義します。

### 検索オプション

検索時間に返すトランザクションには、各イベントのローテキスト、共有イベントタイプ、フィールド値が含まれます。また、トランザクションには、duration および transactiontype フィールドに保存された追加データも含まれます。

- duration には、トランザクションの長さ (最初のタイムスタンプとトランザクションの最後のイベントとの差) が格納されています。
- transactiontype には、トランザクションの名前 (トランザクションのスタンプ名によって transactiontypes.conf で定義されている) が格納されています。

トランザクションはあらゆる検索に追加できます。最高の検索性能を得るには、検索を作成して、トランザクションコマンドへパイプします。

以下のオプションで transaction コマンドを使用します。**注記:** いくつかの transaction オプションは、他の機能と連動しません。

fields=<quoted comma-separated list of fields>

- 設定した場合、各イベントは、同じトランザクションの一部とみなされる同じフィールドを持つ必要があります。
- 複数フィールドは引用符を使って指定します。(例: fields="field1, field2")
- 共有のフィールド名を持ち、異なる値を持つイベントは、グループ化されません。
  - ◆ 例えば、fields=host のとき、検索結果に host=mylaptop がある場合は、検索結果が </code>host=myserver</code> となるため、同じトランザクションとみなされません。
  - ◆ 検索結果にホスト値がない場合は、host=mylaptop を持つ結果のトランザクションとなることがあります。
- **注記:** 1 つ以上のフィールドを指定する場合は、以下のように、すべてのフィールドを引用符で囲んでください。  
transaction fields="host,thread"

match=closest

- トランザクション定義で使用する照合タイプを指定します
- 現在サポートされている値は、最も近い値のみです。

maxspan=[<integer> s|m|h|d]

- トランザクション内のイベント間を一時停止する最大値を設定します。
- 秒、分、時間、日数で指定できます。
  - ◆ 例: 5s、6m、12h、30d
- デフォルトは 2s (秒) です

maxpause=[<integer> s|m|h|d]

- トランザクション間を一時停止する最大値を指定します。
- トランザクションのイベント間に maxpause より大きい値の一時停止しないようにすることを必要とします。
- 負の値を指定した場合は、maxpause の制約は無効となります。
- デフォルトの maxpause は、2 秒です。

startswith=<string>

- トランザクションを開始するために true となる SQLite 表現を指定します。
- 文字列は必ず " " で囲みます。
- SQLite ワイルドカード (%) および単一引用符 ( ' ' ) を使って文字列を指定します。
- この構文は、イベントタイプ名を参照します。(イベント文字列は参照しない)

endswith=<quoted string>

- トランザクションを終了するために true となる SQLite 表現を指定します。
- 文字列は必ず " " で囲みます。
- SQLite ワイルドカード (%) および単一引用符 ( ' ' ) を使って文字列を指定します。
- この構文は、イベントタイプ名を参照します。(イベント文字列は参照しない)

#### トランザクションとマクロ検索

トランザクションとマクロ検索は、トランザクション検索の代わりとなる強力な組み合わせです。トランザクション検索を作成してから、\$field\$ を付けて保存して置換を可能にします。

マクロ検索については、本書の「マクロ検索の設計」を参照してください。

#### トランザクション検索の例

**ある一定の時間内にひとりのユーザー(またはクライアント IP アドレス)が検索したすべてのウェブページをグループ化する検索を実行します。**

この検索は、アクセスログからイベントを抽出し、(3 時間の間に) 双方で 5 分以内に発生した同じ clientip 値を共有するイベントでトランザクションを作成します。

```
sourcetype=access_combined | transaction fields=clientip maxpause=5m maxspan=3h
```

## トランザクションの定義

#### トランザクションの定義

一連のイベントは、トランザクションタイプに変換できます。使用例については、本書の「トランザクションについて」をお読みください。

transactiontypes.conf でトランザクションタイプを作成できます。下の設定詳細を参照してください。

設定ファイルの全般的な内容については、管理者マニュアルの「設定ファイルについて」を参照してください。

#### transactiontypes.conf によるトランザクションタイプの設定

1. \$SPLUNK\_HOME/etc/system/local/、または \$SPLUNK\_HOME/etc/apps/ の独自のカスタムアプリケーションディレクトリに transactiontypes.conf ファイルを作成します。
2. スタンザを作成し、そのスタンザ内の各トランザクションの仕様を一覧してトランザクションを定義します。以下の属性を使用します。

```
[<transactiontype>]
maxspan = [<integer> s|m|h|d]
maxpause = [<integer> s|m|h|d]
fields = <comma-separated list of fields>
exclusive = <true | false>
match = closest
```

```
[<TRANSACTIONTYPE>]
```

- イベントタイプはいくつでも作成できます。それぞれがスタンザ名および複数の以下の属性/値ペアで表されます。
- スタンザ名 [ <TRANSACTIONTYPE> ] を使って、Splunk Web のトランザクションを検索します。
- 以下の属性にエントリーを指定しない場合は、Splunk がデフォルト値を使用します。

```
maxspan=[<integer> s|m|h|d]
```

- トランザクションに対する最大時間長を設定します。
- 秒、分、時間、日数で指定できます。
- ◆ 例： 5s、6m、12h、30d
- デフォルトは 5m(分) です。

```
maxpause=[<integer> s|m|h|d]
```

- トランザクション内のイベント間を一時停止する最大値を設定します。
- 秒、分、時間、日数で指定できます。
- ◆ 例： 5s、6m、12h、30d
- デフォルトは 2s(秒) です。

```
fields = <comma-separated list of fields>
```

- 設定した場合、各イベントは、同じトランザクションの一部とみなされる同じフィールドを持つ必要があります。
- デフォルトは "" です。

```
exclusive = <true | false>
```

- イベントが複数のトランザクションにある、または 1 つのトランザクションを「独占」するかどうかを切り替えます。
- (上述の) 'fields' に適用します。
- 例えば、fields=url,cookie および exclusive=false の場合、'cookie' を持つが'url' 値が異なるイベントが、同じ 'cookie' を共有するが異なる URL を持つ複数のトランザクションにある可能性があります。
- exclusive = false を設定すると、各イベントに対して複数の照合を探すため、処理時間がおおよそ倍になります。
- デフォルトは " true" です。

```
match = closest
```

- 使用する照合タイプを指定します。
- 現在サポートされているのは、"closest" のみです。
- デフォルトは "closest" です。

3. Splunk Web のトランザクションコマンドを使って定義したトランザクションを(トランザクションタイプ名で)呼び出します。検索中に設定仕様を上書きできます。

トランザクションの検索については、本書の「トランザクションの検索」を参照してください。

# 保存済み検索と検索ジョブの管理

## 保存済み検索の管理

保存済み検索の管理

### 思案中

検索の保存およびその共有の基本的な概要については、ユーザーマニュアルの「検索の保存と検索結果の共有」を参照してください。

ここでは、管理で保存済み検索ページの使用を含めて、ナレッジ管理の観点からみた保存済み検索について説明します。

## マクロ検索の設計

マクロ検索の設計

保存済み検索を実行するときに設定する変数であるマクロフィールドを含む保存済み検索を作成します。Splunk Web または Splunk の CLI でマクロ検索を実行できます。

マクロ検索は、検索と似ていますが、グラフィックインターフェイスがないところが異なります。

マクロ検索の設定

1. 保存済み検索を作成します。\$TERM\$ を使って置換用のマクロフィールドを指定します。保存済み検索には、複数のマクロフィールドを含めることができます。

```
host=swan OR host=pearl $user$ $trans$
```

2. 検索に名前を付けて保存します。ここでは、検索を **usertrans** の名前で保存します。
3. ここでマクロ検索を作成します。これは、保存済み検索を呼び出す検索で、保存済み検索のマクロフィールドの変数を特定します。savedsearch 検索コマンドを使用して保存済み検索を呼び出します。その後、保存済み検索で特定したマクロフィールドに値を入力します。キー値ペアを指定して、抽出したフィールド、イベントタイプ、データのその他の値などを検索します。

下の例では、**usertrans** 検索を呼び出し、\$user\$ および \$trans\$ マクロフィールドの値を指定しています。

```
...| savedsearch usertrans user=KateAusten trans=query
```

**注記：** コマンドの前に "|" (パイプ) 演算子を使用します。

上述のマクロ検索は、この検索と同等です。

```
host=swan OR host=pearl user=KateAusten trans=query
```

# フォーム検索の設計

## フォーム検索の設計

フォーム検索は、特定の検索の作成でユーザーをガイドする簡単な検索インタフェースです。これには、以下の機能が含まれます。

- 具体的なフィールド値を持つフィールド (ユーザー名や ID 番号など) を開く。デフォルト値を表示することも可能。
- 動的に定義された検索条件の収集を含む詳細リストの表示
- 特定のフィールド値 ("404"、"500"、"503" などのエラーコード) の選択を強制するラジオボタンの表示
- 1 つのフォームから取得した値を表示する複数の結果パネル。さまざまな隠れた検索に関連付けて、異なるチャートおよびレポートを生成する。

フォーム検索は、Splunk のダッシュボードの構成に使用されるものと同様の XML コードで作成されています。詳しくは、デベロPPER マニュアルの「フォーム検索の構築」を参照してください。

## 保存済み検索とレポートのナビゲーションの定義

### 保存済み検索とレポートのナビゲーションの定義

ナレッジマネージャは、簡単な検索を助長する論理的な方法で、保存済み検索およびレポートが、Splunk アプリケーションの最上位のナビゲーションメニューに表示されるようにしなければなりません。そうするには、使用するアプリケーションに対応するようナビゲーションメニューをカスタマイズする必要があります。ナビゲーションメニューに注意を払わないと、保存済み検索やレポートは後続のカテゴリ化を行わずに追加されるため、時間とともにメニューが長くなり、非効率的になる可能性があります。

アプリケーションに適したトップレベルのナビゲーションメニューで検索を保存し整理する方法を管理するには、ナビゲーションメニュー裏にあるコードを操作する必要があります。コードを操作する場合は、ナビゲーションコードは検索およびレポートのリストを *収集* として参照していることに注意が必要です。

次のトピックでは、保存済み検索とレポートのリストをトップレベルのナビゲーションメニューで管理するために出来ることについて説明しています。ナビゲーションメニューの XML コードの調整の仕方については、デベロPPER マニュアルの「ナビゲーションメニューのカスタマイズ」を参照してください。

### デフォルト収集の設定

各アプリケーションには、「未分類」検索用に設定されたデフォルト収集があります。未分類検索とは、ナビゲーションメニューコードで明確に特定されていない検索を示します。これは、すべての新しく保存された検索にも適用される収集です。例えば、検索 app では、デフォルト収集は **検索とレポート** です。

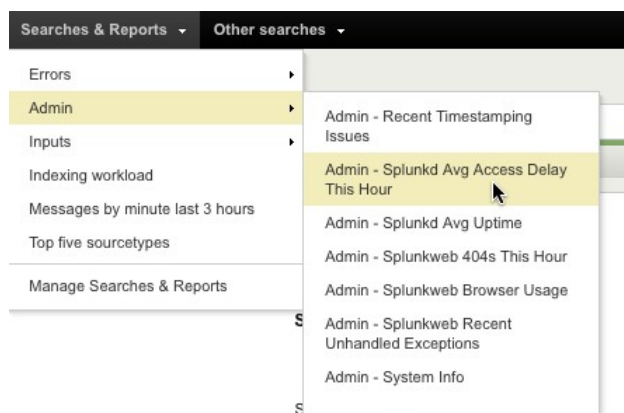
デフォルト収集を設定しない場合は、アプリケーションのトップレベルのナビゲーションメニューに表示されるよう保存済み検索を手動でナビゲーションコードに追加しなければなりません。

**注記：** デフォルト収集は、未分類のビューおよびダッシュボードに対しても設定する必要があります。

## 保存済み検索収集のネスト化

保存済み検索とレポートの数は、アプリケーションの実行と共に増大します。そのため、論理的な方法で検索を整理する方法を見つけることが重要です。手動で、収集を機能別にグループ化する構造を作ることができます。さらには、大きな収集を小さな収集にグループ分けする収集のネスト化を設定することもできます。

検索 app で、収集のネスト化を使って、同類の検索タイプをグループ化します。



## 保存済み検索の動的なグループ化

収集は、名前のサブストリングが一致する保存済み検索を動的にグループ化できるよう設定できます。例えば、上述の検索 app では、すべての未分類検索をタイトルに "admin" 文字を付けて収集のネスト化でグループにしました。

この保存済み検索をサブストリングの照合で動的にグループ化するには、2つの方法があります。

カテゴリー化されていないサブストリング照合検索の収集として、つまり、手動で他の収集に追加されていない検索のみを表示する収集を作成します。

すべてのサブストリング照合検索の収集として、つまり、ナビゲーションメニューのどこに表示されるかに関係なく、サブストリングが一致するすべての検索を表示するコレクションを作成します。

**注記：** いずれの場合も、ナビゲーションメニューに関連付けられているそのアプリケーションで利用可能な保存済み検索とレポートのみが表示されます。

# サマリーインデックスの設定

## サマリーインデックスの設定

### サマリーインデックスの設定

サマリーインデックスの概要、および Splunk Web 使ってサマリーインデックスを設定する方法については、ユーザーマニュアルの「サマリーインデックスを使ってレポートの効率を上げる」を参照してください。

検索で、保存、スケジュール、サマリーインデックスの有効化のアラートオプションを選択しない限り、`savedsearches.conf` の検索用サマリーインデックスを手動で設定することはできません。

このステップを Splunk Web で実施するとき、検索用のサマリーインデックスを有効にしてあると、システムがインデックスを生成します。インデックスは、保存済み検索と同じ名前が付けられます。この時点で、保存済み検索用のサマリーインデックスを手動で設定できます。

検索の保存、スケジュールリング、アラートの設定については、ユーザーマニュアルの「検索を保存して検索結果を共有する」、「保存検索のスケジュールリング」、および「予約検索に対するアラート条件の設定」を参照してください。

**注記：** インデックスの作成に使う検索を定義するとき、ほとんどの場合に、サマリーインデックスの作成に使用する検索のサマリーインデックスレポートコマンドを使用してください。これらのコマンドは、文頭に "si-" が付く `sichart`、`sitimechart`、`sistats`、`sitop`、`sirare` などです。これらのコマンドを使って作成した検索は、最終的に完全なサマリーインデックスのクエリに使用する検索バージョンとなります。

サマリーインデックスのレポートコマンドは、ポピュレート検索の短期間範囲のスケジュールリングや大量のサンプルを採取するポピュレート検索の設定など、下の「サマリーインデックス検索定義の注意事項」に記載される問題を自動的に考慮します。これらの問題は、インデックスの作成に使う検索にサマリーインデックスのレポートコマンドを使用しない場合にのみ、考慮する必要があります。

サマリーインデックスのレポートコマンドを使用しない場合は、予め作成したサマリーインデックスに値を入力する `addinfo` および `collect` 検索コマンドを使って、Splunk が保存およびスケジュールする検索を作成します。この方法については、このトピックの「手動によるサマリーインデックスの入力」を参照してください。

**注記：** サマリーインデックスにインデックス付けするイベントは、ライセンスボリュームに不利になります。本当に必要のない限り、サマリーインデックスに、大量のイベントをインデックス付けしないようにしてください。ライセンスボリュームへの影響については、Splunk サポートにご相談ください。

### 保存済み、スケジュール済み検索のサマリーインデックスのカスタマイズ

Splunk Web を使って、保存済み、スケジュール済み、サマリーインデックス有効検索のサマリーインデックスを有効にすると、Splunk は、スタンプを `$SPLUNK_HOME/etc/system/local/savedsearches.conf` に自動生成します。このスタンプを編集して検索用のサマリーインデックスをカスタマイズできます。

Splunk Web を使って検索を保存およびスケジュールしても、Splunk Web を使って検索用のサマリーインデックスを有効にしている場合、新しく入力するインデックスがある限り、`savedsearches.conf` を使って保存済み検索用のサマリーインデックスを簡単に有効にできます。手動でインデックスを設定する方法については、管理者マニュアルの「インデックスの管理について」を参照してください。

```
[ <name> ]
action.summary_index = 0 | 1
action.summary_index._name = <index>
action.summary_index.<field> = <value>
```

- [`<name>`]: Splunk は、サマリーインデックスが有効になっている保存済みおよびスケジュールした検索の名前を基にスタンザに名前を付けます。
- `action.summary_index = 0 | 1`: 1 と設定してサマリーインデックスを有効にします。0 と設定してサマリーインデックスを無効にします。
- `action.summary_index._name = <index>` - 検索で入力されたサマリーインデックスの名前を表示します。この検索に特定のサマリーインデックスを作成した場合は、ここに名前を入力します。
- `action.summary_index.<field> = <value>`: フィールド/値ペアを指定して、サマリーインデックスにインデックスされた各検索結果に追加します。

**注記:** このフィールド/値ペアは、検索を実行して、イベントデータを入力する際に、サマリーインデックスに含まれるイベントの特定を簡単にする「タグ」の一種として作動します。このキーは、任意ですが、絶対にフィールド/値ペアを1つも持たないサマリーインデックスを設定しないよう推奨しています。

#### サマリーインデックスに便利な検索コマンド

サマリーインデックスは、Splunk Web のインタフェースまたはサマリーインデックスのレポートコマンドを使わずに手動でサマリーインデックスを作成する場合に必要な一連の専用レポートコマンドを活用しています。

- `addinfo`: サマリーインデックスは、`addinfo` コマンドを使って、現在の検索に関する全般的な情報を持つフィールドを、サマリーインデックスに投入される検索結果に追加します。| `addinfo` を任意の検索に追加すると、サマリーインデックスでインデックスされるとどのような結果が得られるか見ることができます。
- `collect`: サマリーインデックスは、`collect` を使って検索結果をサマリーインデックスにインデックスします。| `collect` を使うと、任意の検索結果を別のインデックスにインデックスします(`collect` コマンドオプションを使う)。
- `overlap`: `overlap` を使って、サマリーインデックスの格差と重複を特定します。`overlap` は、サマリーインデックス内でタイムスタンプ値が重複する同じ `query_id` のイベントを検索、またはイベントが欠けている時間的な期間を特定します。

#### サマリーインデックスに投入する検索を手動で設定する

Splunk Web の検索オプションダイアログおよびサマリーインデックスのレポートコマンドを使わずにサマリーインデックスを設定する場合、まず、`indexes.conf` で別のインデックスを設定するようにサマリーインデックスを設定する必要があります。手動でインデックスを設定する方法については、本書の「インデックスの管理について」を参照してください。

**重要:** `indexes.conf` に加えた変更を有効にするには、Splunk を再起動する必要があります。

1. 結果をまとめたい検索を Splunk Web の検索バーから実行します。

- 検索の時間範囲を必ず制限してください。検索で生成される結果の数は、検索用に設定した検索結果限界の最大値を超えないようにする必要があります。
- データに適用するタイムインターバル(10分、2時間、1日など)を必ず選択してください。(Splunk Web のインターバル設定については、ユーザーマニュアルの「保存検索のスケジューリング」を参照してください。)

2. `addinfo` 検索コマンドを使用します。 `| addinfo` を検索の最後に追加します。

- このコマンドは、サマリーインデックスに投入するために、`collect` コマンドで必要とするイベントに、検索に関する情報を追加します。
- 常に `| addinfo` を任意の検索に追加して、サマリーインデックスで検索結果がどのように見えるかプレビューします。

3. `collect` 検索コマンドを追加します。 `|collect index=<index_name> addtime marker="info_search_name=\"<summary_search_name>\""` を検索の最後に付加します。

- `index_name` をサマリーインデックスの名前で置換します。
- `summary_search_name` をこの検索結果をインデックスで見つけるためのキーと置換します。
- `overlap` 検索コマンドを使用してイベントを生成する場合は、`summary_search_name *must*` を設定します。

**注記:** 通常は、提供されている `summary_index` アラートアクションを使用するようにしてください。`addinfo` および `collect` を使った設定には、スケジュール済み検索でサマリーインデックスイベントを生成するときに必要なとしないいくつかの冗長手順が必要です。既に通過した時間範囲に対するサマリーインデックスを逆埋めする場合に手動による設定が必要です。

#### サマリーインデックス検索定義の注意事項

何らかの理由で、サマリーインデックスのレポートコマンドを使わずに、サマリーインデックスのポピュレート検索を設定する場合は、少し時間をかけて処理方法を計画してください。サマリーインデックスでは、鶏の先に卵が来ます。サマリーインデックスの投入に使用する検索の定義を助けるため、実際にレポートしたい検索を使用します。

多くのサマリー検索には、集合統計が関与します。例えば、メインインデックスに毎日数百件ものイベントが増加する中、前日1日のファイアウォール違反に関連する上位10個のIPアドレスの検索をレポートします。

サマリーインデックスで実行した同じ検索の結果をサマリーインデックスに投入すると、統計的に不正確な結果を得る可能性が高くなります。サマリーインデックスに投入する検索を定義するときは、これらのルールに従ってサマリーインデックス検索から生成された集合統計の精度を向上させてください。

#### ポピュレート検索の短時間スケジューリング

サマリーインデックスに投入する検索は、( 頻繁に実行されるため ) インデックスに対して最終的に実行する検索の時間より短い間隔でスケジュールしてください。可能な限り短い時間範囲を設定してください。例えば、毎日「トップ」レポートを作成する必要がある場合は、サマリーインデックスに投入するレポートは 1 時間を基本的にサンプルを採取します。

#### 大量のサンプルを採取するポピュレート検索の設定

サマリーインデックスをに投入する検索では、サマリーインデックスで実行する検索よりも大量のサンプルを検索してください。例えば、不正IPアドレスの上位 10 件を毎日サマリーインデックスで検索する計画がある場合、不正IPアドレスの時間別上位100件をサマリーインデックスに投入する検索を設定します。

この方法には、( 全体的なサンプル収集がより大量および頻繁に行われるため ) 上位 10 件レポートで統計的に精度の高い結果が得られる、上位 20 件または 30 件の不正 IP アドレスのレポートに変更する場合に柔軟性があるという 2 つの利点があります。

サマリーインデックスのレポートコマンドは、完全なサマリーインデックスのクエリを実行する検索より大きなサンプルを自動的に採取します。そのため、正確なイベントデータでサマリーインデックスを作成します。このコマンドを使用しない場合は、`head` コマンドを使って、サマリーインデックスで実行する検索より大量のサマリーインデックスポピュレート検索のサンプルを選択します。つまり、時間別のサマリーインデックスポピュレート検索には `| head=100` を使い、完全なサマリーインデックスの日次検索には `| head=10` を使います。

#### 加重平均を得る検索の設定

サマリーインデックスポピュレーティング検索で平均を出し、サマリーインデックスのレポートコマンドを使用しない場合は、加重平均を得る検索を設定する必要があります。

例えば、時間別、日次、週次で平均応答時間のレポートを作成するとします。これを行うには、「時間平均」で平均して「日間平均」を生成します。残念ながら、日間平均は、各「時間平均」のイベント数が同じでない場合は、正確になりません。加重平均機能を使うと、正しい「日間平均」を得ることができます。

下の表現は、`stats` および `eval` コマンドを `sum` 統計アグリゲータと併用して、加重平均で日間平均応答時間を正確に算出します。この例では、`eval` コマンドが平均応答時間数で合計平均応答時間を分割した結果となる `daily_average` フィールドを生成します。

```
| stats sum(hourly_resp_time_sum) as resp_time_sum, sum(hourly_resp_time_count) as resp_time_count  
| eval daily_average= resp_time_sum/resp_time_count | .....
```

#### ポピュレート検索をスケジューリングしてデータの格差および重複を防ぐ

上述の 2 つのルールに加えて、データ格差および重複を最小限にするには、サマリーインデックスに投入する検索のスケジュールのインターバルおよび遅延を確実に設定します。

サマリーインデックスのデータの格差は、サマリーインデックスでイベントにインデックスを付けられない場合の時間です。この格差は、以下の場合に発生する可能性があります。

- splunkd で失敗した
- 予約保存済み検索 (サマリーインデックス付き) の実行に時間がかかり、次の予約実行時間を過ぎても実行している。例えば、通常実行に 7 分かかる検索に、5 分ごとにサマリーにデータを投入する検索をスケジューリングしたら、前の検索が終わらないと次の検索を実行できないため、問題が発生します。

重複は、同じタイムスタンプを共有するサマリーインデックス (同じ検索) のイベントです。重複イベントは、サマリーインデックスで作成したレポートおよび統計を混乱させます。重複は、保存検索で設定した時間範囲が検索のスケジュールの頻度より長くなる、または collect コマンドを使って手動でサマリーインデックスを実行すると発生する場合があります。

#### サマリーインデックス設定の例

この例では、savedsearches.conf に表示されるウェブ統計のサマリーインデックスの設定を示しています。下に一覧されるキーは、保存済み検索「MonthlyWebstatsReport」のサマリーインデックスを有効にして、サマリーインデックスに投入される各イベントに 2008 の値を持つ Webstatsreport フィールドを付加します。

```
#name of the saved search = Apache Method Summary
[Apache Method Summary]
# sets the search to run at each search interval
counttype = always
# enable the search schedule
enableSched = 1
# search interval in cron notation (this means "every 5 minutes")
schedule = */12****
# id of user for saved search
userid = jsmith
# search string for summary index
search = index=apache_raw startminutesago=30 endminutesago=25 | extract auto=false | stats count by method
# enable summary indexing
action.summary_index = 1
#name of summary index to which search results are added
action.summary_index._name = summary
# add these keys to each event
action.summary_index.report = "count by method"
```

#### サマリーインデックスにより影響を受けるその他の設定ファイル

savedsearches.conf の設定に加えて、indexes.conf および alert\_actions.conf にもサマリーインデックスの設定があります。

Indexes.conf は、サマリーインデックスのインデックス設定を指定します。Alert\_actions.conf は、保存済み検索に関連付けられた警告時の対応 (サマリーインデックスを含む) を制御します。

**注意：** Splunk スタッフの明確な指示がない限り alert\_actions.conf の設定を編集しないでください。