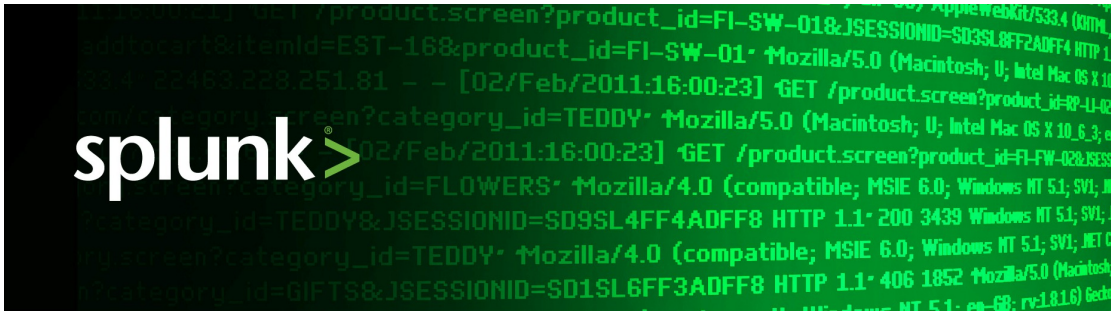


Splunk Enterprise 6.2.0

ダッシュボードと視覚エフェクト

作成：2014 年 11 月 21 日 午後 4 時 10 分

Table of Contents

はじめに

このマニュアルについて

このマニュアルは、Splunk Enterprise を使ったダッシュボードとデータの視覚エフェクトの作成に関する事項を説明しています。

Splunk データの視覚化の概要

- 視覚エフェクトリファレンス – 利用できる各種視覚エフェクトについて (テーブル、グラフ、イベントリスト、その他)。
- 視覚エフェクトのデータ構造要件 – より良いサーチの設計方法を理解するための、各種視覚エフェクトのデータ構造要件の概要。
- 基本的なテーブル/グラフのドリルダウンアクションの概要 – データのドリルダウン方法を学習。

ダッシュボードの作成とメンテナンス

- Splunk Web フレームワーク – Splunk Web フレームワークについて学習。
- ダッシュボードエディタについて – ダッシュボードおよび視覚エフェクトを作成、編集するための、Splunk の対話型エディタについて学習。
- シンプル XML の編集について – シンプル XML を使ったダッシュボードの編集とカスタマイズについて学習。
- ダッシュボード PDF の生成 – ダッシュボードの PDF の生成およびメール配信のスケジュールの学習。

Splunk Enterprise 視覚エフェクトオプション

視覚化リファレンス

Splunk には、サーチ結果を視覚化するためのさまざまなオプションが用意されています。単純な「イベントリスト」視覚エフェクトを使えば、イベントデータを表やグラフ (縦棒、折れ線、面、円など) で表示できます。単一の離散型数値を返すサーチの場合、さまざまなゲージや単一値表示を使って視覚化することができます。

サーチが視覚エフェクトがサポートする構造のデータを返さない場合、視覚エフェクトオプションを制限することができます。たとえば、テーブルおよびグラフ (縦棒、横棒、折れ線、面、円など) 視覚エフェクトの両方をサポートするデータ構造のサーチ結果を返すには、**変換コマンド** (`stats`、`timechart`、または `top`) が必要です。詳細は、このマニュアルの「視覚化のデータ構造要件」を参照してください。

変換コマンドを使ったサーチの作成方法については、『サーチマニュアル』の「変換コマンドとサーチについて」を参照してください。

Splunk 視覚エフェクト定義機能へのアクセス

Splunk には、視覚エフェクトを作成、変更するためのインターフェイスツールが用意されています。これらのツールには、Splunk Web 内のさまざまな場所からアクセスすることができます。

- サーチ
- ダッシュボード
- ダッシュボードエディタ
- ピボット
- レポート

また、シンプル XML コードを直接作成、編集することもできます。

サーチからの視覚エフェクト

[サーチ] ページのサーチ結果の表示方法を変更することができます。サーチの実行後、**[視覚エフェクト]** タブを選択して、視覚エフェクトタイプを選択します。選択した視覚エフェクトの書式設定オプションを指定することができます。サーチは、視覚エフェクトとして書式を設定できる結果を返す、レポートサーチでなければなりません。

「視覚エフェクトの編集」には、ダッシュボード・エディタでの視覚エフェクトの編集に関する説明が記載されています。ダッシュボード・エディタに表示される説明は、[サーチ] ページの視覚エフェクトにも適用されます。

ダッシュボードパネルの視覚エフェクト

サーチ結果を新しいダッシュボードパネルに表示する場合、サーチが返したデータを表すために最適な視覚エフェクトを選択することができます。次に、ビジュアルエディタを使ってパネルへのビジュアル表示をきめ細かく調整することができます。

サーチ結果からダッシュボードパネルを作成するには、サーチの実行後に **[名前を付けて保存]** > **[ダッシュボードに追加]** をクリックします。ダッシュボードの作成と編集の詳細は、「ダッシュボードエディタについて」および「視覚エフェクトの編集」を参照してください。

ダッシュボードエディタ

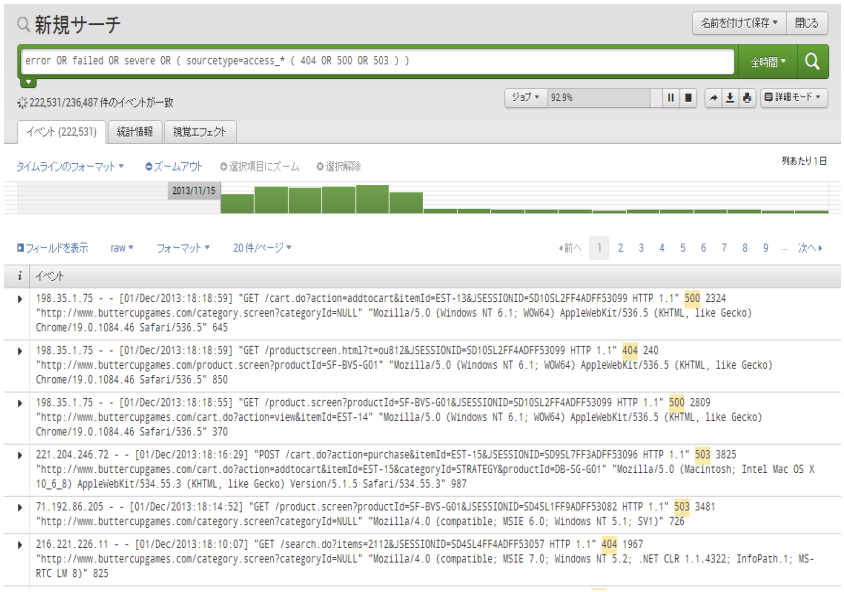
対話型ビジュアルエディタのダッシュボードエディタを使って、視覚エフェクトを作成、編集することができます。詳細は、「ダッシュボードエディタについて」を参照してください。

イベントの視覚エフェクト

イベント視覚エフェクトは、基本的に未加工のイベントをリストに表示したものになります。

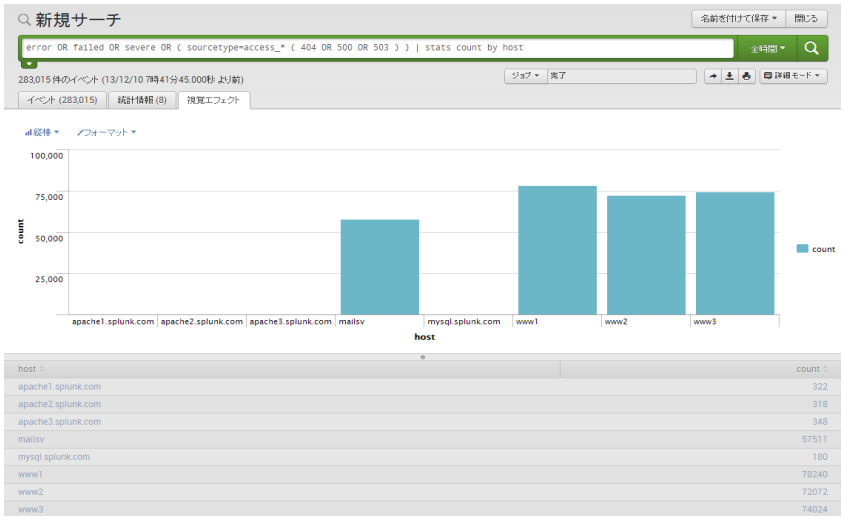
イベント視覚エフェクトは、変換操作を含まない任意のサーチ (例：`stats`、`chart`、`timechart`、`top`、`rare` などのレポート・コマンドを使用するサーチ) で利用できます。たとえば、一連の単語およびフィールド値をサーチすると、イベントのリストが返されます。

```
error OR failed OR severe OR ( sourcetype=access_* ( 404 OR 500 OR 503 ) )
```



このサーチに変換コマンドを追加すると、テーブルやグラフとして表示できる統計的結果を得られます。

```
error OR failed OR severe OR ( sourcetype=access_* ( 404 OR 500 OR 503 ) ) | stats count by host
```



イベントリストでは、以下の作業を行えます。

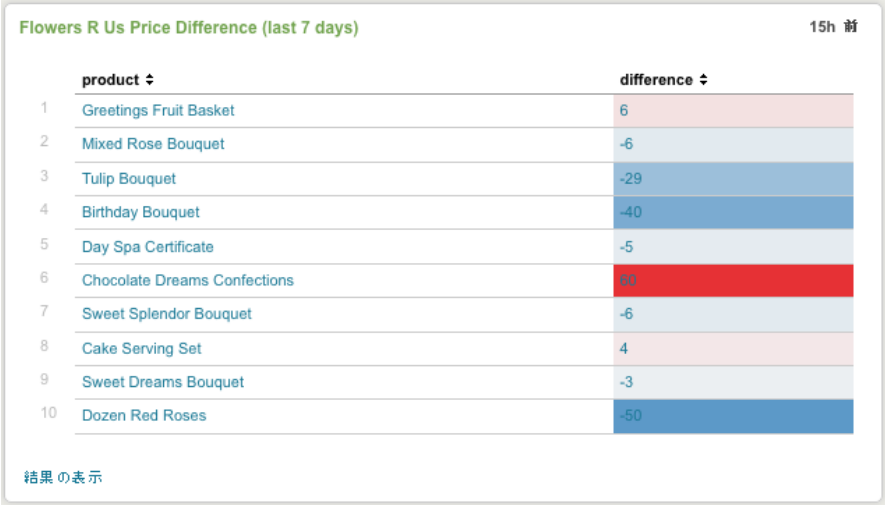
- 表示イベント数を指定する。
- 各イベントの左に数字を表示するかどうかを指定する。
- ページ (またはダッシュボードパネル) 内に収まるようにイベントテキストの折り返しを設定する。

テーブル

テーブル視覚エフェクトは、任意のサーチから生成できます。ただし、`stats`、`chart`、および `timechart` などの変換操作を含むサーチの場合は、より興味深いテーブルが生成されます。

以下の例は、仮想の花屋さんのテーブルを表しています。このテーブルは、そこの商品と仮想の競合他社との料金の違いを追跡しています。以下のサーチは、テーブルのデータを生成します。

```
sourcetype=access_* | stats values(product_name) as product by price, flowersrus_price | eval difference = price - flowersrus_price | table product, difference
```



difference (違い) 列のセルは、色を違えて表示しています。このテーブルは、データオーバーレイのため、ヒートマップを使用しています。値が高い場合は赤で、低い場合は青で表示しています。この例で、競合他社より料金が高い商品は赤で、安い商品は青で表しています。

テーブル視覚エフェクトを使って、以下の作業を行えます。

- 表示するテーブル行数を設定する。
- 行番号を表示する。
- ヒートマップや高/低値インジケータなどのビジュアル情報を提供するデータのオーバーレイを追加する。

ビジュアル・エディタでダッシュボード内のテーブルに書式設定する場合は、**ドリルダウン**機能を設定することができます。行またはセルによるドリルダウンを有効にしたり、テーブル全体のドリルダウンを無効にしたりできます。このマニュアルの「基本的なテーブル/グラフのドリルダウン・アクションの概要」を参照してください。

テーブルのスパークライン

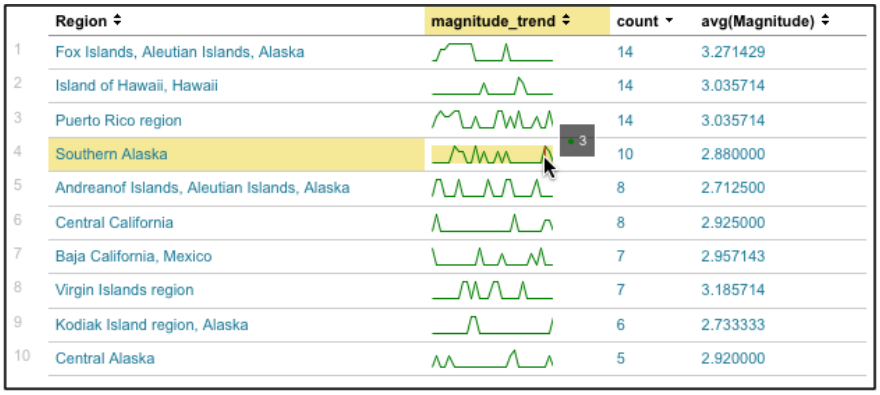
スパークラインを表示するように、テーブル視覚エフェクトを設定することができます。スパークラインは、テーブル結果だけでは判別しにくいデータの隠されたパターンを描き出します。レポートやダッシュボード内のテーブルの有益性と情報密度を向上することができます。

スパークラインを使用するには、サーチ内で変換コマンド `stats` または `chart` を使用する必要があります。これらのコマンドに `sparklines` 関数を追加して、テーブルにスパークライン列を追加します。詳細は、『サーチ・マニュアル』の「サーチ結果へのスパークラインの追加」を参照してください。

USGS 自身データを調査するこのサーチによるスパークラインの例を以下に示します。**USGS Earthquake Feeds** から最新の CSV ファイルをダウンロードして、それを Splunk への入力として追加することができます。ただし、フィールド名とフォーマットが、ここの例とは異なる場合があります。この例では、特定の 7 日間に記録された、マグニチュード 2.5 以上のすべての地震を表しています。

```
source=usgs | stats sparkline(avg(Magnitude),6h) as magnitude_trend, count, avg(Magnitude) by Region | sort count
```

このサーチでは、当該期間中に地域当たりに発生した地震数合計が上位 10 件の地域を表示します。結果テーブルのスパークラインは、各地域で当該期間に発生した地震のマグニチュードの傾向を表しています。またこの例では、スパークライン上にマウスカーソルをかざすと、その地点での値が表示されることも分かります。



グラフ

Splunk は、縦棒、折れ線、面、散布図、円グラフなど、さまざまなグラフ視覚エフェクトが用意されています。これらの視覚エフェクトには、結果に 1 つ以上の**シリーズ**を含む変換操作を指定したサーチが必要です。

シリーズは、グラフに描画できる一連の関連データポイントです。たとえば、折れ線グラフに描画される各折れ線は、それぞれが別個のシリーズを表しています。単一のシリーズを生成するサーチを作成することも、複数シリーズを持つデータを返すサーチを作成することも可能です。

変換サーチが生成するテーブルを考えてみましょう。テーブル内の最初の列以降の各列が、それぞれ異なるシリーズを表しています。「単一シリーズ」サーチは、2 列のみのテーブルを生成します。「複数シリーズ」サーチは、3 列以上の列を持つテーブルを生成します。

グラフ視覚エフェクトは、単一シリーズサーチを表示できます。ただし、横棒、縦棒、折れ線、および円グラフ視覚エフェクトは一般的に最善のデータを表示します。円グラフは単一シリーズサーチのデータのみを表示できます。

サーチが複数のシリーズを生成する場合、横棒、縦棒、折れ線、面グラフ、および散布図は、最善のデータを表示します。

詳細は、このマニュアルの「視覚化のデータ構造要件」を参照してください。

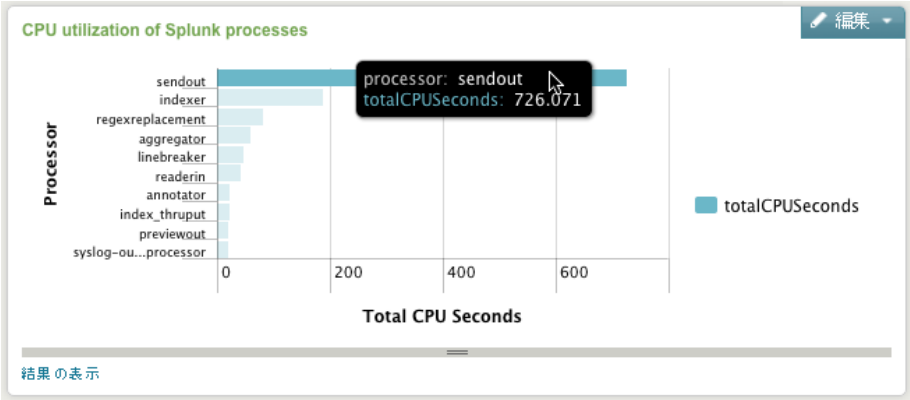
縦棒および横棒グラフ

データ内のフィールド値の頻度を比較するには、縦棒または横棒グラフを使用します。縦棒グラフの場合、一般的に X 軸の値はフィールド値になります。`timechart` 変換コマンドを使用するサーチの場合、X 軸は時間を表します。Y 軸には任意のフィールド値、値カウント、またはフィールド値の統計計算を表示できます。縦棒グラフと横棒グラフはデータを同じように表していますが、X 軸と Y 軸の値が逆になります。詳細は、このマニュアルの「視覚化のデータ構造要件」を参照してください。

以下の横棒グラフは、内部指標を使ってこのサーチの結果を表しています。過去 15 分のプロセッサ別 CPU 秒の合計を算出します。次に、プロセッサ上位 10 件の合計を降順に表示します。この例では、横棒または縦棒グラフでそのバー上にカーソルを移動して詳細情報を表示できることも分かります。

以下のサーチは横棒グラフ視覚エフェクトを使用しています。

```
index=_internal "group=pipeline" | stats sum(cpu_seconds) as totalCPUSeconds by processor | sort 10 totalCPUSeconds
desc
```



縦棒および横棒視覚エフェクトを使って、以下の作業を行えます。

- グラフのタイトル、および X 軸と Y 軸のタイトルを設定する。
- 最小の Y 軸値を設定する。
- 単位を対数に設定する。
  非常に小さな Y 軸値と非常に大きな Y 軸値が混在している場合に、対数値が役立ちます。詳細は、このマニュアルの「視覚エフェクトの編集」を参照してください。
- グラフのスタック、100% スタック、およびスタックなしを指定する。
  デフォルトでは、横棒および縦棒グラフはスタックなしになります。スタックされた横棒/縦棒グラフの詳細は、以降のサブセクションを参照してください。
- Y 軸の主軸単位を設定する。
  たとえば、データに最適な目盛りの単位を設定します。
- グラフの凡例の位置、および凡例ラベルの省略方法を設定する。
- **ドリル ダウン**機能を有効化/無効化する。
  このマニュアルの「基本的なテーブル/グラフのドリルダウン・アクションの概要」を参照してください。

スタックした (積み上げ) 縦棒および横棒グラフ

ベースサーチに複数の**データシリーズ**が存在する場合、スタックした (積み上げ) 縦棒/横棒グラフを使って、データ内のフィールド値の頻度を比較することができます。

スタックなしグラフ

スタックしない縦棒グラフの場合、各シリーズの縦棒が相互に隣接して表示されます。スタックなし縦棒グラフは、比較的単純なサーチ結果に役立ちます。しかし、シリーズ数が増えると、スタックなし縦棒グラフは乱雑で分かりにくくなってしまいます。

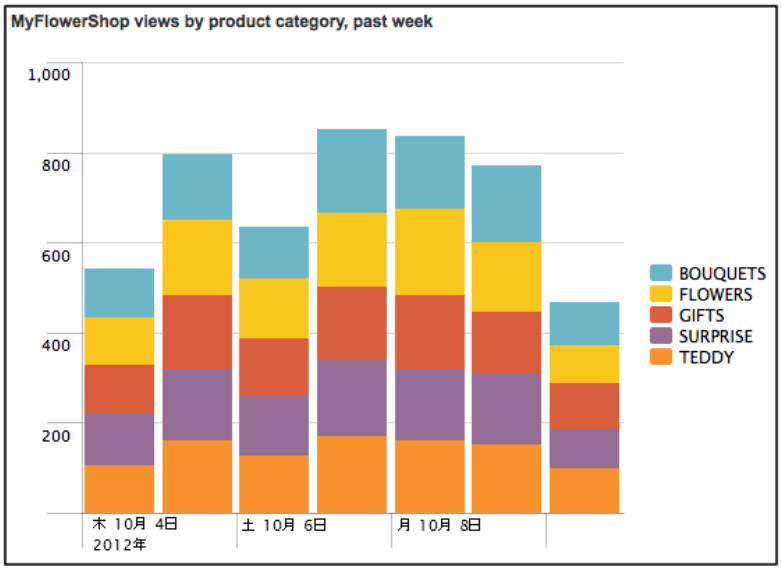
スタックグラフ

スタック縦棒グラフは、単一のデータポイントのすべてのシリーズの列を、単一の列のセグメントとして表示します。列の合計値は、各セグメントの合計になります。一般的にスタックされた縦棒/横棒グラフを使って、特定のデータセットを構成する異なる種類のデータの相対的な重みまたは重要度を強調表示することができます。

以下の例は、仮想の花屋さんの Web サイトのページを参照した顧客を表しています。ページビューは、7 日間に渡って製品カテゴリ別に表示されています。

この例のデータは、以下のサーチで取得されています。サーチ内で `fields` コマンドを使用することにより、グラフには製品カテゴリ ID を持つイベントの数のみが表示されます。サーチ結果内では `null` に分類される、カテゴリ ID を持たないイベントは除外されます。

```
sourcetype=access_* method=GET | timechart count by categoryId | fields _time BOUQUETS FLOWERS GIFTS SURPRISE TEDDY
```



100 パーセントスタックグラフ

!00% スタックが設定されたグラフでは、縦棒または横棒グラフ内のデータ分布を、縦棒または横棒のサイズの割合で比較することができます。縦棒または横棒内の各データセグメントが、利用可能なすべてのデータに対する割合を表しています。

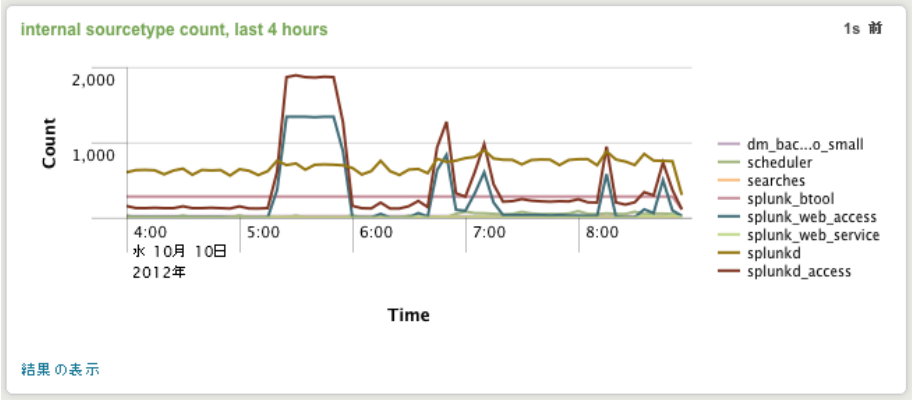
[100% スタック] は、極めて小さなスタックと極めて大きなセグメントが混在している場合に、横棒/縦棒グラフの各セグメント間のデータ分布をより詳細かつ明確に把握するために役立ちます。

折れ線および面グラフ

一般的に折れ線および面グラフは、時間の推移に伴うデータの傾向を表すために用いられます。ただし、X 軸を使って時間以外の任意のフィールド値を表すことができます。グラフに複数のシリーズがある場合、異なる色で折れ線や面が表示されます。

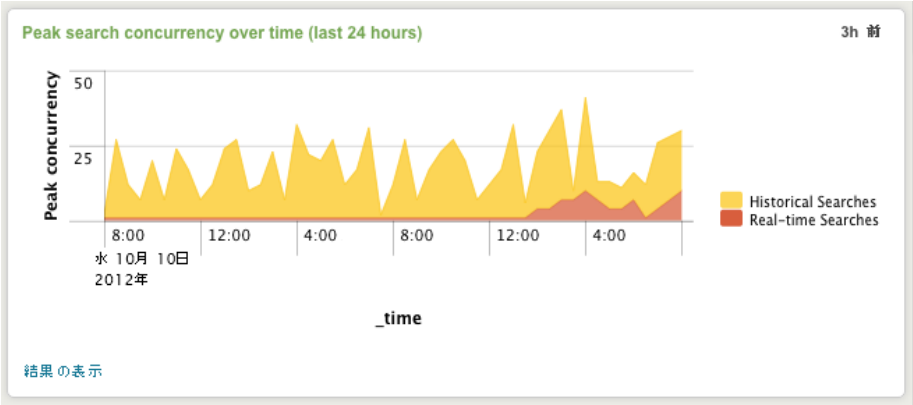
例の折れ線グラフを生成するサーチを以下に示します。

```
index=_internal | timechart count by sourcetype
```



面グラフの影付き領域は、数量を強調するために役立ちます。この例の面グラフは、以下のサーチで取得されています。

```
index=_internal source=*metrics.log group=search_concurrency "system total" NOT user=* | timechart
max(active_hist_searches) as "Historical Searches" max(active_realtime_searches) as "Real-time Searches"
```



折れ線および面グラフでは以下の作業を行えます。

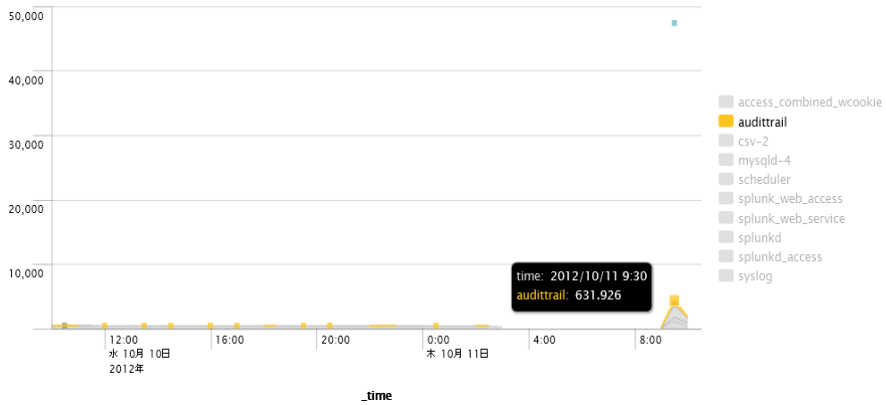
- グラフのタイトル、および X 軸と Y 軸のタイトルを設定する。
- NULL のY 軸値の表示方法を指定する。
  NULL のデータポイントのギャップを放置する、0 のデータポイントに接続する、または次の正のデータポイントに接続することができます。ギャップを放置する場合、グラフには接続されていないデータポイント用のマーカーが表示されます。この場合、他の正のデータ・ポイントとは隣接しません。
- 最小の Y 軸値を設定する。
- 単位を対数に設定する。
  非常に小さな Y 軸値と非常に大きな Y 軸値が混在している場合に、対数値が役立ちます。詳細は、このマニュアルの「視覚エフェクトの編集」を参照してください。
- Y 軸の主要単位を設定する。
  たとえば、データを最適に表す単位に目盛を設定します。
- グラフの凡例の位置、および凡例ラベルの省略方法を設定する。
- ドリル ダウン機能を有効化/無効化する。
  ドリルダウンの詳細は、このマニュアルの「基本的なテーブル/グラフのドリルダウンアクションの概要」を参照してください。

スタックされた折れ線および面グラフ

スタック折れ線および面グラフは、スタック縦棒および横棒グラフと似ています。スタックされた折れ線/面グラフは、複数のシリーズが存在するグラフを参照するユーザーに、データセット全体に対する各データシリーズの関係を手軽に把握できる手段を提供しています。

このサーチは、スタック面グラフをベースにしています。この例は、データポイント上にマウスを移動した際の情報の表示も表しています。

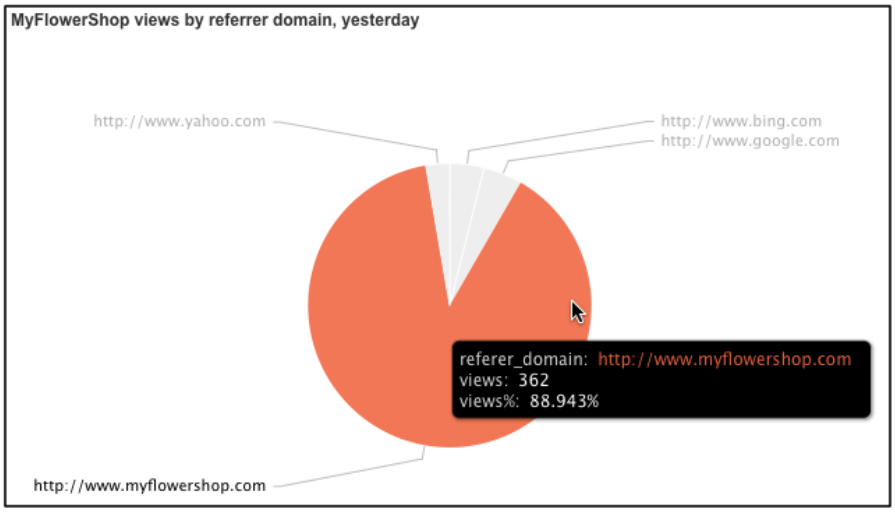
```
index=_internal per_sourcetype_thruput | timechart sum(kb) by series useother=f
```



円グラフ

全体的なデータセットに対する一部のデータの関係を表す場合に、円グラフを使用します。円グラフのスライスのサイズは、すべての値の合計に対する当該データ値の割合として表されます。

以下の円グラフは、仮想のオンラインストアへの参照元ドメインによる前日の参照数を表しています。円グラフの個別のスライス上にマウスカーソルを移動すると、詳細が表示されます。



円グラフのプロパティを定義する際に、グラフのタイトルを設定することができます。ビジュアルエディタでダッシュボード内の円グラフの書式設定を行っている場合は、以下の作業を行えます。

- グラフのタイトルを設定する。
- グラフの凡例の位置を設定する。
- **ドリル ダウン**機能を有効化/無効化する。
  ドリルダウンの詳細は、このマニュアルの「基本的なテーブル/グラフのドリルダウンアクションの概要」を参照してください。

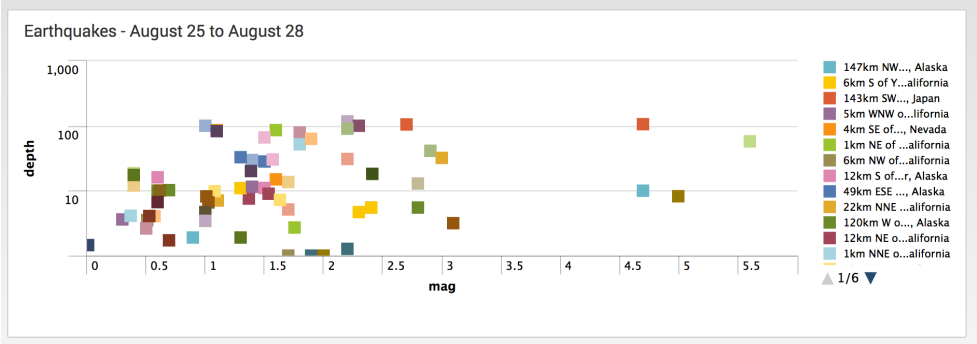
散布図

散布図は、データの離散値間の傾向を表示する場合に使用します。一般的に散布図は、定期的には発生しない、またはシリーズに所属しない離散値を表します。これは、定期的なポイントのシリーズを描画する折れ線グラフとは異なります。

以下の例は、USGS 地震データを使った散布図を表しています。データは、過去 30 日間に記録されたすべての地震のデータを含む CSV ファイルから取得します。

この例のサーチは、特定の 3 日間に発生した地震のマグニチュードと深度を描画します。散布図の点は、地震の場所を表しています。この散布図の例は、以下のサーチで生成されています。

```
index=usgs_earthquake place="*" earliest=1408950000 latest=1409295600 | table place mag depth
```



散布図に必要なデータ構造の詳細は、このマニュアルの「データ構造の視覚化要件」を参照してください。

散布図では、以下の作業を行えます。

- グラフのタイトル、および X 軸と Y 軸のタイトルを設定する。
- NULL の Y 軸値の表示方法を指定する。
  NULL のデータ・ポイントのギャップを放置する、0 のデータ・ポイントに接続する、または次の正のデータ・ポイントに接続することができます。ギャップを放置する場合、グラフには接続されていないデータポイント用のマーカーが表示されます。この場合、他の正のデータ・ポイントとは隣接しません。
- 最小の Y 軸値を設定する。
- 単位を対数に設定する。
  非常に小さな Y 軸値と非常に大きな Y 軸値が混在している場合に、対数値が役立ちます。詳細は、このマニュアルの「視覚エフェクトの編集」を参照してください。
- Y 軸の主要単位を設定する。
  たとえば、データを最適に表す単位に目盛を設定します。
- グラフの凡例の位置、および凡例ラベルの省略方法を設定する。
- **ドリル ダウン**機能を有効化/無効化する。
  ドリルダウンの詳細は、このマニュアルの「基本的なテーブル/グラフのドリルダウンアクションの概要」を参照してください。

バブル・チャート

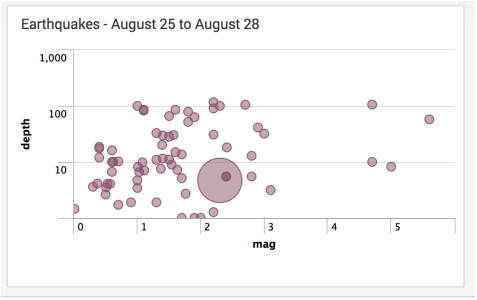
バブル・チャートは、3 次元のシリーズを表示する手段を提供しています。各ポイント (バブル) は、グラフの X 軸と Y 軸の 2 次元に対して描画されます。バブルのサイズが、3 番目の次元の値を表しています。

3 日間の地震データを表したバブル・チャートの例を以下に示します。X 軸と Y 軸は、記録された地震のマグニチュードと深度を表しています。

このバブル・チャートの例は、以下のサーチで生成されています。

```
index=usgs_earthquake place="*" earliest=1408950000 latest=1409295600 | stats count by place, mag, depth
```

バブルのサイズは、見つかった地震数を表しています。大きなバブル上にマウス・カーソルを移動すると、そのマグニチュードと深度に対してカウントが 2 であることが分かります。その他のバブルのカウントは 1 です。マウス・カーソルを移動すると、シリーズからのその他のデータ (地震の場所) も表示されます。



Viz bubble example2.png

バブル・チャートでは、以下の作業を行えます。

- バブルの最低/最大サイズを設定する。
- バブルのサイズを面積または直径で設定する。
- グラフのタイトル、および X 軸と Y 軸のタイトルを設定する。
- NULL の Y 軸値の表示方法を指定する。
  NULL のデータ・ポイントのギャップを放置する、0 のデータ・ポイントに接続する、または次の正のデータ・ポイントに接続することができます。ギャップを放置する場合、グラフには接続されていないデータポ

イント用のマーカーが表示されます。この場合、他の正のデータ・ポイントとは隣接しません。

- 最小の Y 軸値を設定する。
- 単位を対数に設定する。
  非常に小さな Y 軸値と非常に大きな Y 軸値が混在している場合に、対数値が役立ちます。詳細は、このマニュアルの「視覚エフェクトの編集」を参照してください。
- Y 軸の主要単位を設定する。
  たとえば、データを最適に表す単位に目盛を設定します。
- グラフの凡例の位置、および凡例ラベルの省略方法を設定する。
- **ドリル ダウン**機能を有効化/無効化する。
  ドリルダウンの詳細は、このマニュアルの「基本的なテーブル/グラフのドリルダウンアクションの概要」を参照してください。

単一値視覚エフェクト

単一値およびゲージは、単一値を返す変換サーチの結果を表示します。たとえば、特定のサーチ基準セットのイベント数合計を返すイベントを考えてみましょう。以下のサーチは、Splunk Enterprise インスタンスの過去 1 時間のエラー数合計を返します。

```
index=_internal source="*splunkd.log" log_level="error" | stats count as errors
```

サーチが単一値を返すようにする方法はいろいろとあります。たとえば、`top` コマンドと `head=1` を組み合わせます。

> **警告：**単一値を返すサーチの使用には注意が必要です。ダッシュボードエディタでダッシュボードの視覚エフェクトを設計する場合、サーチが複数の値を返す場合でも単一値視覚エフェクトを選択することができます。この場合、単一値視覚エフェクトは結果テーブルの最初のセルの値を使用しますが、それが目的のセルではない可能性もあります。

単一値視覚エフェクトのデータ構造要件の詳細は、このマニュアルの「視覚化のデータ構造要件」を参照してください。

単一値視覚エフェクト

単一値視覚エフェクトには、単一の数値を返すサーチの結果が表示されます。単一値を返すリアルタイムサーチの場合、サーチにデータが到着するにつれて表示される数値が変化します。

単一値を表示する視覚エフェクトで、返された値が定義された範囲のどれに該当するかによって、色を変更するように設定できます。範囲を定義するには、`rangemap` サーチコマンドを使用します。パネルエディタを使って、単一値視覚エフェクトの範囲マップを設定することもできます。デフォルトで単一値視覚エフェクトは、以下の範囲マップ設定を使用します。

- `low`：緑
- `elevated`：黄色
- `severe`：赤

以下のサーチは、上記の単一値視覚エフェクトを使用しています。

```
index=_internal source="*splunkd.log" log_level="error" | stats count as errors | rangemap field=errors low=0-3
elevated=4-20 default=severe
```

単一値視覚エフェクトを使って、以下の作業を行えます。

- パネルタイトルを指定する。
- 結果の前、後、下にテキストを指定する。
- **ドリル ダウン**機能を有効化/無効化する。
  ドリルダウンの詳細は、このマニュアルの「基本的なテーブル/グラフのドリルダウンアクションの概要」を参照してください。

ゲージについて

Splunk には、放射状、フィラー、およびマーカーの 3 種類のゲージが用意されています。

ゲージは、単一の数値を、特定の意味または論理を持つ一定範囲の色とマップします。ゲージは、単一値視覚エフェクトで説明した範囲マップを使って、色の範囲を定義しています。時間の経過に伴い値が変化すると、ゲージマーカーの位置がその範囲内で変化します。ゲージは、特に**リアルタイムサーチ**向けに、動的な視覚エフェクトを提供しています。返される値の変動に伴い、ゲージマーカーが一定範囲内で前後に移動します。

以下の各種ゲージの例は、同じベースサーチを使用しています。

```
index=_internal source="*splunkd.log" log_level="error" | stats count as errors
```

放射状ゲージ

放射状ゲージは、速度計や圧力計に似た外観となっています。これには、円弧状の範囲目盛と回転する針があります。単一値視覚エフェクトで説明した範囲マップを使って、フィラー・ゲージの色範囲を定義します。

針の現在値がゲージの下部に表示されます。下の例では、値が 19 になっています。値の範囲が指定された最小値を下回ったまたは最大値を超えた場合、針が上限 (または下限) の境界で振動します。

以下の例は「補完」と「最小」版の放射状ゲージを表しています。

4.3 radial gauge minimal-1.png

フィラーゲージ

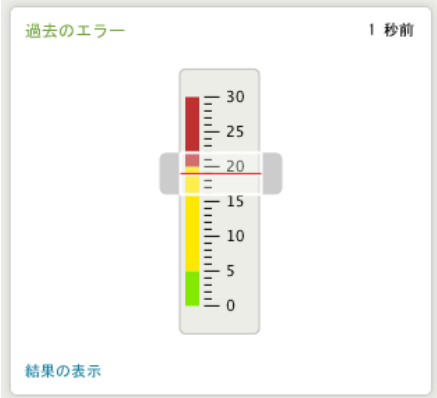
フィラーゲージは温度計と似た外観で、液体状のフィラーインジケータがゲージの範囲境界を越えて増加すると、その色が変化します。単一値視覚エフェクトで説明した範囲マップを使って、フィラー・ゲージの表示色を定義します。

デフォルトで、フィラーゲージは垂直に表示されます。フィラーゲージを水平に表示するように設定できます。

マーカーゲージ

マーカー・ゲージは、すでに塗られているフィラー・ゲージの線形バージョンです。サーチが返された値の位置に、ゲージのマーカーが表示されます。単一値視覚エフェクトで説明した範囲マップを使って、マーカー・ゲージの表示色を定義します。

ゲージにリアルタイムサーチの結果を表示している場合、マーカーは返された値の変動に伴って一定範囲内で前後に移動します。返された値がマーカーゲージの範囲の上限または下限を超えた場合、マーカーはその上限/下限の境界で振動します。



デフォルトで、マーカーゲージは垂直に表示されます。マーカーゲージを水平に表示するように設定できます。

マーカーゲージには、3 桁を超える値の表示時に問題があります。これに対処するには、大きな値を係数で除算して小さな値に変換するようにサーチを設定してください。たとえば、一般的に返される値が 10,000 を超えるような場合に、結果を 1,000 で除算します。こうすることにより、値 19,100 が返された場合に、19.1 として表示できます。

また、範囲をパーセントで返すようにグラフオプションを設定して、大きな数値に対処することもできます。

Splunk Web を使ったゲージ視覚エフェクトの設定

ビジュアルエディタを使って、ダッシュボードパネル内にゲージを設定できます。ビジュアルエディタでは、以下の事項を設定できます。

- パネルのタイトルを指定する。
- ゲージのサイズと値の範囲を定義する。
  たとえば、0 から始まり 100 で終了するゲージを作成し、それに 0～25、26～50、51～75、および 76～100 の範囲を設定することができます。また、1000 から始まり 3000 で終了するゲージを作成し、それに複数の小さな範囲を設定することもできます。
- 各範囲に色を設定する。
  デフォルトでは、最初の 3 つの範囲が緑、黄色、赤になります。色をカスタマイズしたり、範囲を減らしたりすることができます。
- ゲージ・スタイルに「補完」または「最小」を設定する。
  たとえば、「補完」版の放射状ゲージは、現実の機械ゲージと同様の、メタリックなダイヤルと、黒い背景を持っています。「最小」版の放射状ゲージは、余計な装飾を排除した平坦な放射状ゲージ・デザインとなっています。

ビジュアル・エディタでゲージ視覚エフェクトを設定する場合、色の範囲を自動的に定義できます。そのためには、サーチ文字列に定義されている値と gauge コマンドを使用します。ビジュアルエディタが提供するデフォルトの設定をカスタマイズすることができます。

ビジュアルエディタを使ったダッシュボードパネル視覚エフェクトの設定については、このマニュアルの「視覚エフェクトの編集」を参照してください。

その他、レポート・ビルダー、[詳細グラフ] ビュー、およびサーチ App の結果領域を利用して、ゲージ視覚エフェクトを定義することができます。これらの場所では、ゲージ視覚エフェクトのタイトルのみを指定できます。デフォルトでは、以下の 3 つの範囲を持つゲージが作成されます。

- 1～30：緑
- 31～70：黄色
- 71～100：赤

これらの視覚エフェクト定義オプションで異なるゲージ範囲を設定するには、gauge コマンドを使ってサーチを変更する必要があります。

gauge コマンドによるゲージ範囲の設定

gauge コマンドを使って、ゲージ視覚エフェクトのカスタム範囲を設定することができます。

gauge コマンドでは、デフォルトの色を使ってゲージ範囲を設定できます。デフォルトの 3 色は範囲の順番に応じて、緑、黄色、赤となります。gauge を使って、ゲージの対象となるフィールドを指定します。次に、サーチ文字列に範囲の開始と終了を表す範囲値、および各カラーバンドの相対サイズを追加します。

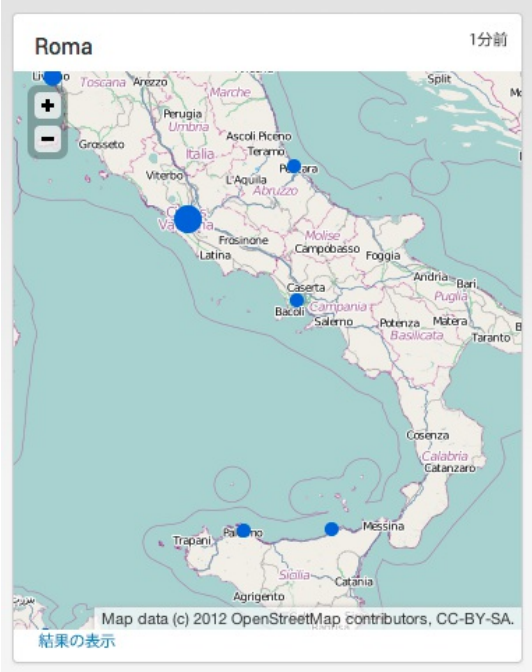
たとえば、範囲が 100～119、120～139、140～159、160～179、および 180～200 の範囲を持つ hitcount 値を追跡するゲージを設定するには、サーチ文字列に以下の項目を追加します。

```
...| gauge hitcount 100 120 140 160 180 200
```

サーチに gauge コマンドを指定しない場合、または指定したけれども範囲を指定していなかった場合、範囲値は次のデフォルト値になります：0 30 70 100。

地図

Splunk Enterprise には、世界地図上に地理的座標を対話型のマーカーとして描画する、地図視覚エフェクトが用意されています。一般的に地図視覚エフェクト用のサーチは、地図上にマーカーを描画するために、`geostats` サーチコマンドを使用します。`geostats` コマンドは、`stats` コマンドと似ていますが、地図用のズームレベルとセルに関するオプションが用意されています。`geostats` コマンドは、マーカー用に緯度と経度の座標を含むイベントを生成します。



他の視覚エフェクトオプション

以下の Splunk 視覚エフェクトは、Splunk Web またはシンプル XML を使って利用することはできません。これらの視覚エフェクトには、アドバンスト XML と Splunk Web フレームワークのモジュールシステムが必要です。

- ヒストグラム
- 範囲マーカーグラフ
- 比率横棒グラフ
- 値マーカーグラフ

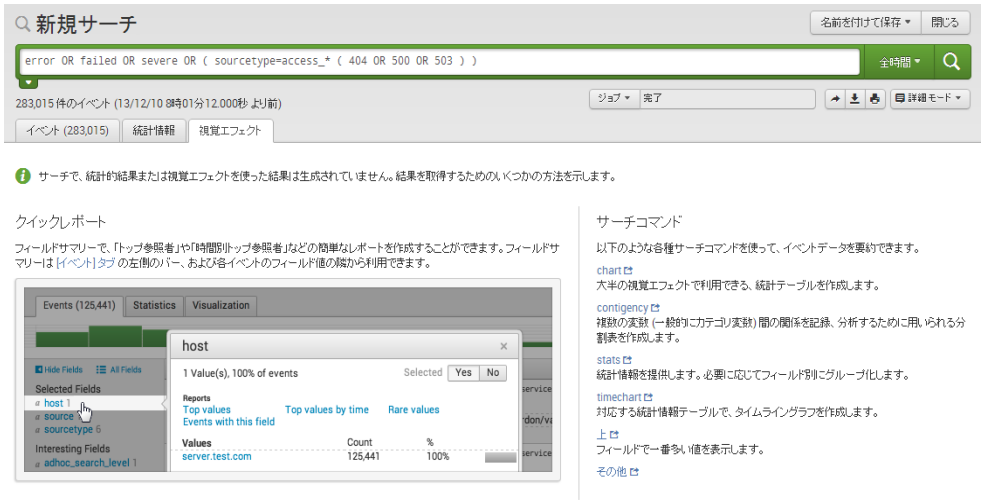
バブルグラフを使って、データ内の離散値の傾向と相対重要度を表すことができます。バブルのサイズは、値の相対重要度を表しています。これは、X 軸と Y 軸上に表示される、バブルによる第 3 の次元を表しています。この次元は、グラフ内の他と相対的なバブルのサイズを決定します。

範囲マーカーグラフと値マーカーグラフは、横棒、縦棒、折れ線、または面グラフ上のオーバーレイとして機能するように設計されています。

これらのグラフタイプ、それらに対して必要なデータ構造、およびそのビュー XML プロパティについては、「カスタムグラフリファレンス」を参照してください。

視覚エフェクトのデータ構造要件

ここでは、利用できる各種視覚エフェクトのデータ構造要件について説明しています。



既存のダッシュボードパネルでベースサーチを変更した時に上記のようなエラーが発生する場合、または新しいパネルの作成時に目的の視覚エフェクトを利用できないような場合、ベースサーチが返すデータがその視覚エフェクトに対応していない可能性があります。たいていの場合、サーチを修正して目的のデータを得ることは簡単です。

たとえば、大部分のグラフ (縦棒グラフ、折れ線グラフ、面グラフ、横棒グラフなど) には、最低 2 つの列を持つテーブル形式の構造のサーチ結果が必要です。最初の列が X 軸値を、それ以降の列がグラフ内の各**シリーズ**に対する Y 軸値を表します (円グラフは単一シリーズの情報のみを提供し、その他のタイプのグラフは複数シリーズを表すことができます)。このようなテーブルを取得するには、ベースサーチに `stats`、`chart`、または `timechart` などのレポートコマンドを設定する必要があります。

- Splunk 視覚エフェクトオプションの概要については、このマニュアルの「視覚化リファレンス」を参照してください。
- ビジュアルエディタを使ったダッシュボードパネルの視覚化については、「視覚エフェクトの編集」を参照してください。
- 視覚エフェクトの設定の詳細は、「グラフ設定リファレンス」を参照してください。

縦棒、折れ線、および面グラフ

縦棒、折れ線、および面グラフは 2 次元グラフで、1 つまたは複数の**シリーズ**をサポートしています。これらのグラフは、デカルト座標系にデータを描画します。データは最低 2 つの列を持つテーブルから取得します。テーブルの最初の列には X 軸値が、以降の列には Y 軸値 (各列がシリーズになる) が含まれます。このことが、「値の推移」サーチや splitby を含むサーチが、縦棒、折れ線、および面グラフとして表示できる理由です。

サーチから縦棒、折れ線、または面グラフを作成する場合、そのサーチは前述の基準を満たすテーブルを生成していなければなりません。たとえば、`timechart` コマンドを使用する任意のサーチが、最初の列が `_time` のテーブルを生成します (それらの結果から、縦棒、折れ線、または面グラフの X 軸が生成される)。レポートコマンドを含む基本サーチの大部分から、同様の結果を得ることができます。

たとえば、`over` 演算子が `source` が X 軸であることを示すこのサーチ：

```
...| chart avg(bytes) over source
```

は、以下のような 2 列の単一シリーズテーブルを生成します。

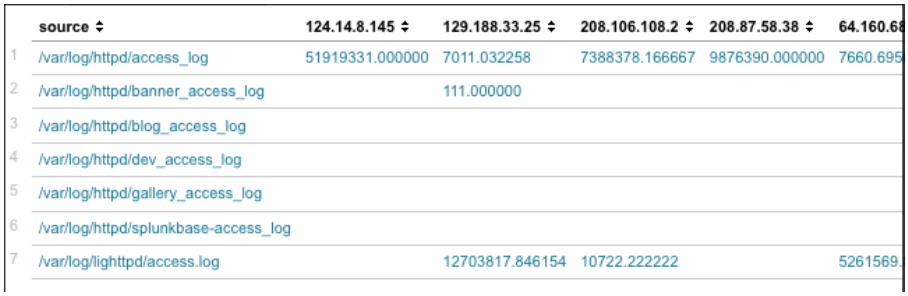
	source	avg(bytes)
1	/var/log/httpd/access_log	57435.562670
2	/var/log/httpd/banner_access_log	686.726415
3	/var/log/httpd/blog_access_log	36833.004373
4	/var/log/httpd/dev_access_log	318.866667
5	/var/log/httpd/gallery_access_log	291.000000
6	/var/log/httpd/splunkbase-access_log	34347.619022
7	/var/log/lighttpd/access.log	121517.319102

このテーブルで、X 軸は `source`、Y 軸は `avg(bytes)` になります。これを使って、各ソースを通過した平均バイト数を比較する縦棒グラフを生成できます。

サーチに `clientip` を splitby フィールドとして追加してみましょう。

```
...| chart avg(bytes) over source by clientip
```

この場合、複数シリーズを持つテーブルが生成されます。



	source	124.14.8.145	129.188.33.25	208.106.108.2	208.87.58.38	64.160.68
1	/var/log/httpd/access_log	51919331.000000	7011.032258	7388378.166667	9876390.000000	7660.695
2	/var/log/httpd/banner_access_log		111.000000			
3	/var/log/httpd/blog_access_log					
4	/var/log/httpd/dev_access_log					
5	/var/log/httpd/gallery_access_log					
6	/var/log/httpd/splunkbase-access_log					
7	/var/log/lighttpd/access.log		12703817.846154	10722.222222		5261569.

このテーブルで X 軸は引き続き `source` で Y 軸も `avg(bytes)` です。ただし、`clientip` で `avg(bytes)` が分割され、複数シリーズのテーブルが作成されます。このデータを表すために、*スタック*縦棒グラフを生成することができます。

有効な X 軸または Y 軸値が欠けている複雑なサーチを作成すると、トラブルが発生してしまいます。このような問題は、たとえば `eval` および `fields` コマンドを使って、完成したテーブル内の列を特定の配置に強制するような場合に発生します。

横棒グラフ

横棒グラフには縦棒、折れ線、面グラフと同じデータ構造要件が適用されます。ただし、X 軸と Y 軸が逆になります。このグラフは 2 つ以上の列を持つテーブルからデータを取得します。テーブルの最初の列には Y 軸値が、以降の列には X 軸値が含まれます。

円グラフ

円グラフは 1 次元で、単一の**シリーズ**のみをサポートしています。このグラフは 2 列のみのテーブルから値を取得します。テーブルの最初の列には円グラフの各スライスのラベルが、2 列目には各ラベルに対応する数値が含まれます。この数値により、各スライスの相対的なサイズが決まります。サーチが生成するテーブルにその他の列が存在する場合、円グラフではそれらの列は何も意味を持たず、無視されます。

前述の 2 つの「縦棒、折れ線、および面グラフ」サーチの例で、最初のもののみが円グラフの作成に利用できます。`source` 列がウェッジのラベルを、`avg(bytes)` 列が各ウェッジの相対サイズを表します (サーチが返す `avg(bytes)` の合計に対する割合)。

散布図

散布図は、データを点在するマーカーで表すデカルトグラフです。これは、各 X 軸値に対して複数の Y 軸値があるような場合に役立ちます (複数シリーズでない場合でも)。データセットは、以下のいずれかの形式になります。

- **単一シリーズ**セットアップ：グラフは 2 列のテーブルから構成され、最初の列 (列 0) には X 軸に描画する値、2 番目の列 (列 1) には Y 軸に描画する値が含まれます。
- **複数シリーズ**セットアップ：グラフは 3 列のデータテーブルから構成されます。最初の列 (列 0) にはシリーズ名が、次の 2 列にはそれぞれ X 軸と Y 軸に描画する値が含まれます。

散布図を生成するには、以下のようなサーチで直接イベントをグラフ化する必要があります。

```
* | fields - _* | fields clientip bytes
```

このサーチは、さまざまなクライアント IP から受信したすべてのパケットを検索し、それを各パケットのバイト数に応じて並べ替えます。

- このサーチでは、`_time` フィールドのような、アンダースコアから始まるすべてのフィールドが削除されることに注意してください。
- 2 番目の `fields` コマンドは、それぞれ X 軸と Y 軸に使用する 2 つのフィールドを分離します。最良の結果を得るために、Y 軸値は数値でなければなりません。(この場合、X 軸は `clientip`、Y 軸は `bytes` になります。)

シンプル XML を使用すれば、ダッシュボード内により複雑な散布図を設定することができます。詳細は、「グラフ設定リファレンス」内の「面、横棒、縦棒、折れ線グラフ、および散布図」および「散布図固有のプロパティ」を参照してください。

ゲージと単一値視覚エフェクト

ゲージと単一値視覚エフェクトは、単一の数値フィールド値を返すサーチを表しています。単一値視覚エフェクトは単に数値を表示しますが、ゲージは定義されている範囲内のどこに値が存在しているのかを表します。

これを利用する例としては、特定の期間またはリアルタイムウィンドウ内でサーチ基準に一致したイベント数を返すサーチが挙げられます。リアルタイムサーチをベースにするゲージの場合、時間の経過とともにリアルタイムサーチウィンドウに表示される値が変化するにつれて、グラフの範囲マーカーも変動します。

同じサーチに対して単一値視覚エフェクトを使用した場合、リアルタイムサーチが返す値の変化に伴って、表示される値が増減します。このサーチと一緒に `rangemap` コマンドを使用した場合、返される値によって単一値視覚エフェクトの色が変化します。

地図

Splunk には、世界地図上に地理的座標を対話型のマーカーとして描画する、地図視覚エフェクトが用意されています。地図視覚エフェクト用のサーチは、地図上にマーカーを描画するために、geostats サーチコマンドを使用する必要があります。geostats コマンドは、stats コマンドと似ていますが、地図用のズームレベルとセルに関する結果を提供します。生成されるイベントには、緯度/経度座標が含まれています。

詳細は、以下の項目を参照してください。

- 『視覚化リファレンス』の「地図」
- 『シンプル XML リファレンス』の「<map> エレメント」
- 『サーチリファレンス』の「Geostats」

ドリルダウン動作

視覚エフェクトでは、デフォルトでドリル ダウン動作が有効になっています。例外は単一値視覚エフェクトで、ドリル ダウン動作はデフォルトで無効になっています。ユーザーが視覚エフェクトをクリックすると、クリックした場所から取得された値で、詳細なサーチが実行されます。詳細なサーチは、[サーチ] ページに表示されます。詳細なサーチはオリジナルのサーチを複製しますが、最後の**変換コマンド**を削除して、それに代わって視覚エフェクトから取得した値を設定します。

ドリルダウンで取得される値は、視覚エフェクトによって異なります。ドリルダウン動作は、パネルエディタまたはベースとなるシンプル XML コードで設定します。パネル・エディタからドリルダウン動作を設定する方法については、「デフォルトのドリルダウン動作」を参照してください。シンプル XML コードでのドリルダウン動作の設定については、『シンプル XML リファレンス』の「パネルの視覚化エレメント」を参照してください。

ドリルダウン動作をカスタマイズして、動的なドリルダウンを実装することができます。動的なドリルダウンを実装するには、シンプル XML コードで <drilldown> タグやその他の関連タグを使用します。動的なドリルダウンでは、結果を表示する他のページへのリンクを指定したり、同じページでの状況に依存したドリルダウンを指定したりすることができます。詳細は動的ドリルダウンの説明を参照してください。

デフォルトのドリルダウン動作

デフォルトのドリルダウン動作は、視覚エフェクトによって異なります。

『シンプル XML リファレンス』には、シンプル XML コードでのドリルダウン動作の設定に関する詳細が記載されています。パネルエディタからドリルダウン動作を設定することもできます。

1. ダッシュボードから、**[編集] > [パネルの編集]** をクリックします。
2. パネルから、視覚エフェクト設定アイコンをクリックします。
3. **[ドリルダウン]** フィールドに、ドリルダウンオプションを指定します。
4. **[適用]** をクリックします。**[完了]** をクリックします。

注意：パネルエディタから、単一値視覚エフェクトに対してドリルダウンを指定することはできません。ドリルダウン動作はシンプル XML コードで指定してください。詳細は単一値視覚エフェクトの説明を参照してください。

以降のトピックでは、各視覚エフェクトのドリルダウンプロパティについて説明していきます。

グラフ

グラフ視覚エフェクトの場合、パネルエディタではドリルダウン動作に関する 2 つのオプションが用意されています。

ドリルダウンオプション	説明
はい	**デフォルト**：ドリル ダウン動作を有効にします。
いいえ	グラフのドリルダウンを無効にします。

グラフのドリルダウン動作は、グラフをクリックしたのか、またはグラフの凡例をクリックしたのかによって異なります。

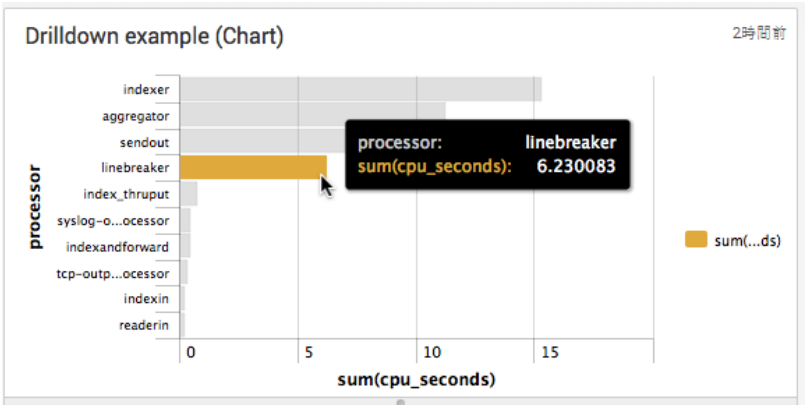
グラフのデータポイントをクリックすると、グラフのY 軸のフィールドまたはシリーズの値を取得して、ドリルダウンサーチが実行されます。例外は Y 軸を持たない円グラフです。円グラフのドリルダウンサーチは、選択されたセグメントの値を取得します。

グラフの凡例をクリックすると、グラフのベースサーチにクリックしたフィールドを追加したドリルダウンサーチが生成されます。凡例表示がフィールドではなく計算された値の場合、ドリルダウンサーチはベースサーチになります。

横棒グラフのドリルダウン例

この例の横棒グラフは、以下のサーチを使って結果を表示します。

```
index="_internal" source="*metrics.log" group="pipeline" | chart sum(cpu_seconds) over processor | sort 10 -
sum(cpu_seconds)
```

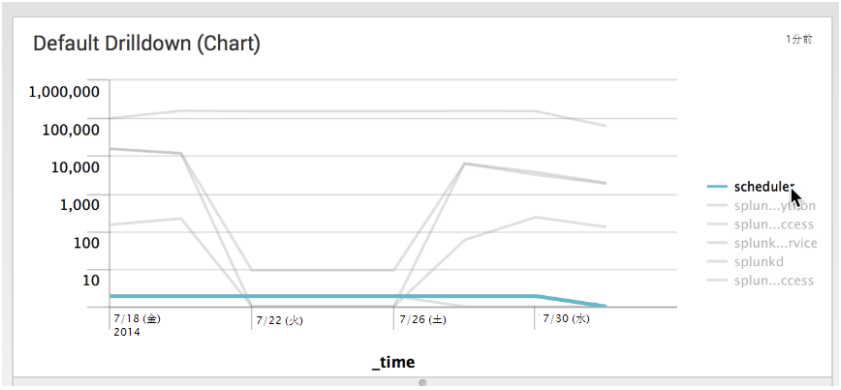


linebreaker プロセッサをドリル・ダウンすると、以下の詳細サーチが生成されます。

```
index="_internal" source="*metrics.log" group="pipeline" processor=linebreaker
```

グラフ凡例のドリルダウン例

この例は、グラフ凡例のフィールドをクリックした時のドリルダウンサーチを表しています。



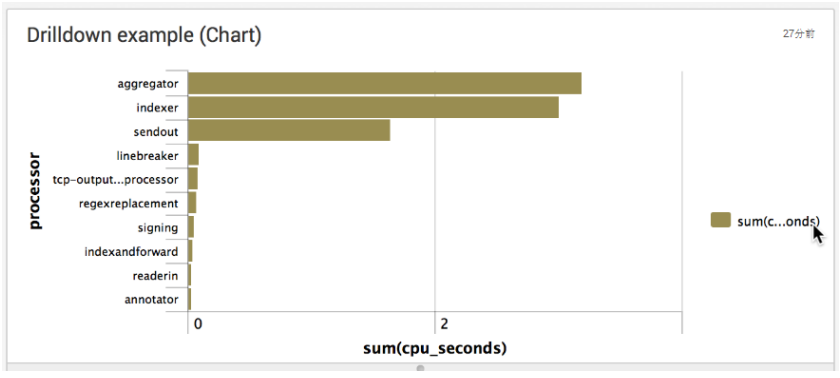
グラフを表示するサーチ：

```
index=_internal | timechart count by sourcetype
```

グラフ凡例の scheduler フィールドをクリックした時のドリルダウンサーチ：

```
index=_internal sourcetype=scheduler
```

この例は、グラフ凡例の計算された値をクリックした時のドリルダウンサーチを表しています。



グラフを表示するサーチ：

```
index="_internal" source="*metrics.log" group="pipeline" | chart sum(cpu_seconds) over processor | sort 10 -
sum(cpu_seconds)
```

グラフ凡例の計算された値をクリックした時のドリルダウンサーチ：

```
index="_internal" source="*metrics.log" group="pipeline"
```

イベントの視覚エフェクト

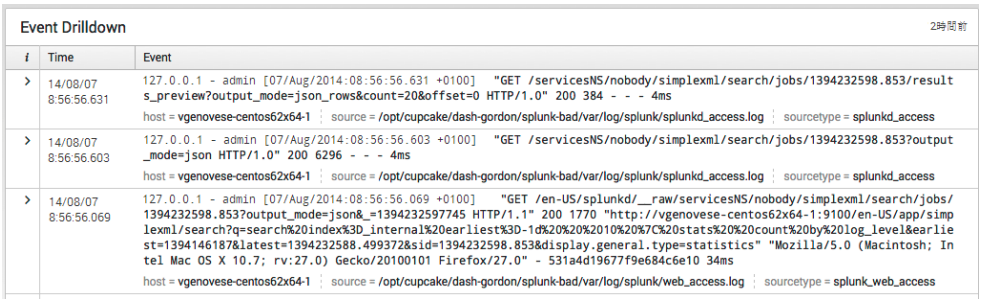
3 種類のイベント表示方法があります。

- raw
- リスト
- テーブル

イベント視覚エフェクトで利用できるドリルダウンオプションは、これらの表示タイプによって異なります。

このセクションの例は、以下のサーチを使用します。以下の画面は、イベントリスト内のサーチを表していますが、データを raw イベントとして、またはテーブル内に表示することもできます。

```
index=_internal earliest=-1d | stats count by log_level
```



イベントは、Splunk Enterprise の中核を為す概念です。イベントの詳細は、「イベントタイプについて」を参照してください。

raw およびリストとしてのイベント

raw イベントとして、またはリスト内に表示されるデータの場合、ドリルダウン動作はマウスカーソルによる、イベントリスト内のセグメントの選択によって異なります。選択のタイプは、完全、内部、または外部として指定できます。「イベントのセグメント分割のタイプ」を参照してください。

ドリルダウンオプションに応じて、マウスカーソルをメジャーセグメント、連続マイナーセグメント、またはマイナーセグメント上に移動します。マウスカーソルを選択項目上に移動したら、クリックして詳細サーチを実行します。

以下の例は、ドリルダウンのためのイベントの選択方法を表しています。この例では、上記の視覚エフェクトからイベントを取得しています。

ドリルダウンオプション	説明	例
完全	メジャーセグメント、または 1 つまたは複数の連続マイナーセグメントを取得します。最初の例は、マイナーセグメント上へのマウスカーソルの移動を表しています。2 番目の例は、メジャーセグメントの選択例を表しています。	"GET /servicesNS/admin/simplexml/data/ui/viewstates?search=name%3)" 200 1868 - - - 8ms Viz drilldownEventFull.png
内部	単一のマイナーセグメントを選択します。	"GET /servicesNS/admin/simplexml/data/ui/viewstates?search=name%3)" 200 1868 - - - 8ms
外部	完全なメジャーセグメントを選択します。	"GET /servicesNS/admin/simplexml/data/ui/viewstates?search=name%3)" 200 1868 - - - 8ms
なし	ドリルダウンを無効にします。	—

テーブルとしてのイベント

イベントをテーブルとして表示した場合、テーブル内のセルを選択してドリルダウンすることができます。これにより、行内の最初の列 (イベントの時間) の値に基づいた、詳細サーチが実行されます。テーブルに表示されているイベントのドリルダウンを有効/無効にすることができます。

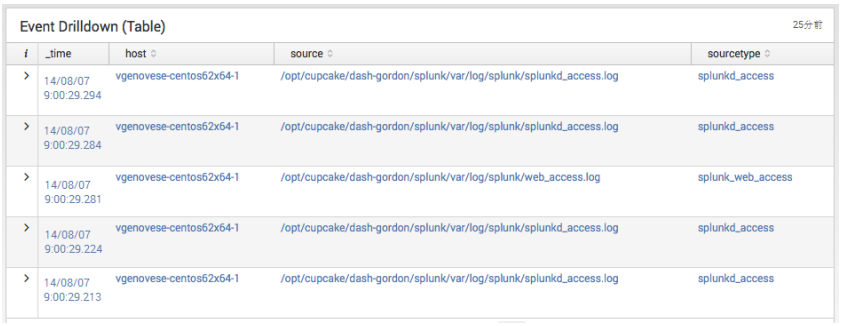
グラフ視覚エフェクトの場合、パネルエディタではドリルダウン動作に関する 2 つのオプションが用意されています。

ドリルダウンオプション	説明

オン	**デフォルト**：ドリルダウン動作を有効にします。
オフ	ドリルダウン動作を無効にします。

以下の視覚エフェクトは、イベントをテーブルとして表示します。任意のセルをクリックしてドリルダウンすることができます。このテーブルは、イベント視覚エフェクトの紹介の例と同じサーチを使用しています。

```
index=_internal earliest=-1d | stats count by log_level
```



結果となる詳細サーチは、サーチの最初の列 (ベースサーチ用に指定された時間) の値を取得します。

```
index=_internal earliest=-1d
```

マップ

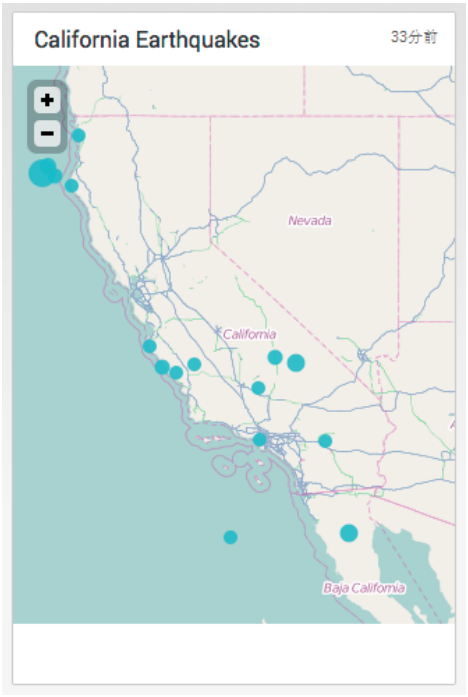
マップ (地図) 視覚エフェクトは、マップ上のクラスタのデフォルトのドリルダウン動作を提供します。クラスタをクリックすると、クラスタの境界に基づいて詳細サーチが生成されます。ドリルダウンに利用できるすべてのマップトークンの詳細は、「マップイベントトークン」を参照してください。

マップ視覚エフェクトには、ドリルダウン動作用の 2 つのオプションがあります。

ドリルダウンオプション	説明
はい	**デフォルト**：ドリルダウン動作を有効にします。
いいえ	ドリルダウンを無効にします。

以下のサーチは、過去 30 日間にカリフォルニアで発生したマグニチュード 3 を超える地震の地図を生成します。

```
index=main mag>3 | geostats latfield=latitude longfield=longitude count
```



地震データを示すクラスタをクリックすると、クラスタ境界の緯度/経度に基づいて詳細サーチが生成されます。

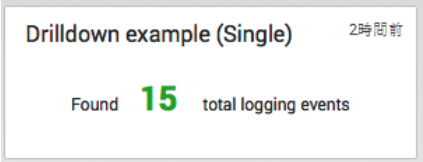
```
index=main mag>3 | search latitude>=36.21094 latitude<36.56250 longitude>=-122.34375 longitude<-121.64062
```

注意：この例では、USGS Earthquakes Web サイトからダウンロードした地震データを使用しています。

単一値

単一値視覚エフェクトのドリルダウン動作は、シンプル XML コード内で有効にします。ドリルダウンオプションには **all** を指定します。

```
<single>
  <searchString>
    index=_internal source="*splunkd.log" (log_level=ERROR
    OR log_level=WARN* OR log_level=FATAL
    OR log_level=CRITICAL) | stats count as log_events
    | rangemap field=log_events low=1-100 elevated=101-300 default=severe
  </searchString>
  <title>Log events</title>
  <earliestTime>-1d</earliestTime>
  <latestTime>now</latestTime>
  <option name="classField">range</option>
  <option name="afterLabel">total logging events</option>
  <option name="beforeLabel">Found</option>
  <option name="drilldown">all</option>
</single>
```



視覚エフェクト内の値をクリックすると、以下の詳細サーチが生成されます。

```
index=_internal source="*splunkd.log" (log_level=ERROR OR log_level=WARN* OR log_level=FATAL OR log_level=CRITICAL)
```

テーブル

テーブル視覚エフェクトには、ドリルダウン動作用の 3 つのオプションがあります。

ドリルダウンオプション	説明
セル	**デフォルト**：選択したセルに対して、行の最初の列の値および選択した列の値を取得します。生成されるドリルダウンサーチは、これらの値をサーチします。
行	ドリルダウンサーチ用に、選択した行内のすべてのセルから値を取得します。
なし	テーブルのドリルダウンを無効にします。

この例は、以下のテーブルから派生した、**[行]** および **[セル]** ドリルダウンオプションを表しています。このテーブルに表示されるデータは、以下のサーチで取得されています。

```
index=_internal earliest=-1d | stats count by sourcetype log_level component
```

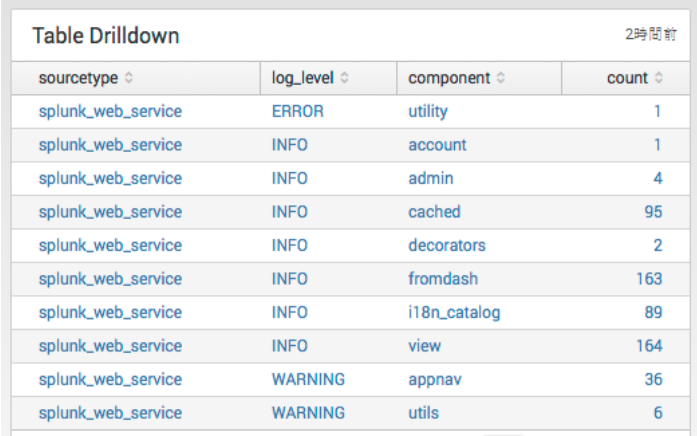


Table Drilldown 2時間前

sourcetype	log_level	component	count
splunk_web_service	ERROR	utility	1
splunk_web_service	INFO	account	1
splunk_web_service	INFO	admin	4
splunk_web_service	INFO	cached	95
splunk_web_service	INFO	decorators	2
splunk_web_service	INFO	fromdash	163
splunk_web_service	INFO	i18n_catalog	89
splunk_web_service	INFO	view	164
splunk_web_service	WARNING	appnav	36
splunk_web_service	WARNING	utils	6

[行] ドリルダウンオプション

[行] ドリルダウンオプションでテーブルを設定し、テーブル内のセルをクリックすると、ドリルダウンサーチはテーブル内のすべての列の値を使用します。この例で、最初の行の任意の場所をクリックすると、以下のドリルダウンサーチが生成されます。

```
index=_internal earliest=-1d sourcetype=splunk_web_service log_level=ERROR component=utility
```

[セル] ドリルダウンオプション

[セル] ドリルダウンオプションでテーブルを設定し、テーブル内のセルをクリックすると、選択したセルの列の値と行内の最初の列の値を組み合わせたドリルダウンサーチが生成されます。この例で、最初の行の **[log_level]** 列をクリックすると、以下のサーチが生成されます。

```
index=_internal earliest=-1d sourcetype=splunk_web_service component=utility
```

動的ドリルダウン

ドリルダウン動作をカスタマイズするには、動的ドリルダウンを使用します。動的ドリルダウンにより、生成される詳細サーチに対して以下のカスタムターゲットを指定することができます。

- Splunk Enterprise 内の App のダッシュボードまたはフォーム
- サードパーティの URL
- 同じページ内の場所 (状況依存のドリルダウン)

動的ドリルダウンエレメント

動的ドリルダウンは、シンプル XML コード内に、<drilldown> エレメントとその他のシンプル XML エレメントを使って実装します。詳細は、『シンプル XML リファレンス』の「ドリルダウンエレメント」を参照してください。

エレメント	説明
<drilldown>	カスタム宛先を定義します。他の動的ドリルダウンエレメントの親エレメント。
<condition>	ドリルダウンアクションを生成するフィールドを指定します。
<link>	詳細サーチのターゲット宛先を指定します。
<set>	ダッシュボード内の他のエレメントやサーチが利用できる、グローバルトークンを公開します。ドリルダウンの結果を同じダッシュボードに表示する場合は、<set> および <unset> を使用します。「状況に応じたドリルダウン・エレメント」を参照してください。
<unset>	前に設定されたトークンを削除します。ドリルダウンの結果を同じダッシュボードに表示する場合は、<set> および <unset> を使用します。ドリルダウンの結果を同じダッシュボードに表示する場合は、<set> および <unset> を使用します。「状況に応じたドリルダウン・エレメント」を参照してください。

ドリルダウンイベントトークン

動的ドリルダウンは、ドリルダウンイベントトークンを使って、視覚エフェクトから取得した値をカスタマイズします。利用できるトークンは、視覚エフェクトによって異なります。このマニュアルの「ダッシュボードでのトークンの使用」および「ドリルダウンのトークンの定義」を参照してください。

たとえば、マップ視覚エフェクトの場合、トークンはマップマーカーのフィールドと値、および緯度/経度の値を示します。テーブル視覚エフェクトの場合、トークンはクリックされたセルから返された名前と値を示します。

テーブル視覚エフェクトで利用できるドリルダウンイベントトークンの一覧を以下の表に示します。すべての視覚エフェクトで利用できるトークンの完全なリストについては、『シンプル XML リファレンス』の「ドリルダウンイベントトークン」を参照してください。

トークン	説明
click.name	テーブルに表示されている一番左のフィールドの名前。存在する場合は、常に `_time` になります。
click.value	クリックされた行の一番左側の列の値。
click.name2	クリックされた列名。
click.value2	クリックされた列の値。
row.<fieldname>	表示されていないフィールドも含めて、クリックされた行のすべてのフィールド値。
earliest/latest	クリックされたテーブル行の時間範囲、または存在しない場合は、サーチの時間範囲。

ドリルダウンイベントトークンは、<set> エレメントで定義するトークンとは異なります。視覚エフェクト内でクリックされた値を取得するために、ドリルダウンイベントトークンは事前定義されています。<set> エレメントを使って定義されたトークンは、ターゲット宛先が使用する値を示します。

宛先リンクの指定

<link> エレメントは、動的ドリルダウンの宛先を指定するための、各種オプションを提供しています。詳細は、『シンプル XML リファレンス』の「<link> エレメント」を参照してください。

以下の事項を指定することができます。

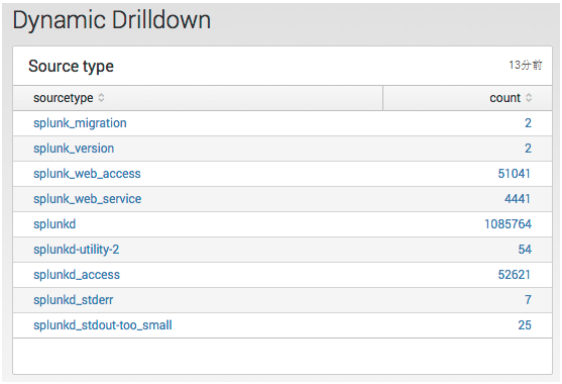
- Splunk Enterprise インスタンス内の同じまたは異なる App 内のダッシュボードを指定する。
- 宛先ターゲット内のフォームに設定する、トークン値を渡す。
- 宛先フォーム内のサーチ用語を定義する、もっとも早い値およびもっとも遅い値を渡す。
- サードパーティの URL を渡す、または必要に応じてドリルダウンアクションが取得した値をクエリ引数として渡す。
- <a> HTTP アンカータグの `target` 値を指定する。これは、ターゲット HTTP Web ページを開く方法を示しています。

<condition> エレメントと一緒に使用すると、ドリルダウンの値を取得するフィールドまたはシリーズの名前を指定することができます。

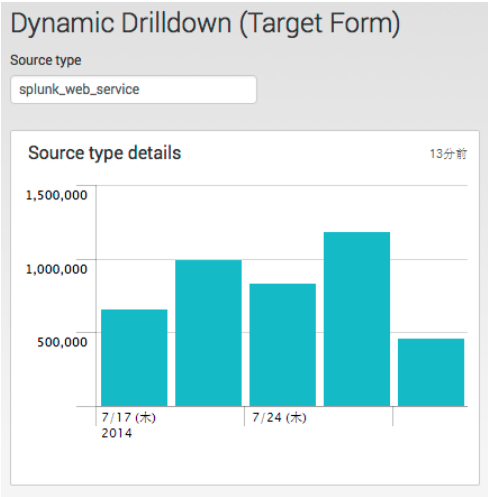
動的ドリルダウンの例

この例は、ダッシュボードから別の App にあるフォームにドリルダウン値を渡す方法を表しています。ダッシュボードにはテーブルが含まれています。テーブルの行内の任意の場所をクリックすると、行内の最初の列からソースタイプの値が取得されます。この値はフォームに入力値として渡されます。

これは、テーブルを含むダッシュボードです。



これは、別の App 内にあるフォームです。ダッシュボードから渡された値が、フォームの入力となります。ダッシュボードのユーザーがソースタイプ splunk_web_service の行内の任意の場所をクリックすると、フォームに結果が表示されます。



動的ドリルダウンを実装したダッシュボード

- <drilldown> および <link> エレメントを使用します。
- <link> に `target` 属性を指定して、ターゲットを新しいページで開きます。
- ターゲットフォームに定義される `src_type_tok` トークンを参照します。
- ドリルダウンオプションに **row** を指定します。

フォーム

- `src_type_tok` トークンを定義します。
- テキスト入力にトークンに対して渡された値を設定して、フォームを実行します。

動的ドリルダウンを実装する、ダッシュボード内のテーブルのソースコード：

```
<dashboard>
  <label>Dynamic Drilldown</label>
  <row>
    <panel>
      <table>
        <search>
          <query>index="_internal" | chart count by sourcetype | sort sourcetype</query>
          <earliest>-7d@h</earliest>
          <latest>now</latest>
        </search>
        <drilldown>
          <link target="_blank">
            /apps/MyApp/drilldown_dynamic_target_form?form.src_type_tok=$row.sourcetype$
          </link>
        </drilldown>
        <option name="drilldown">row</option>
      </table>
    </panel>
  </row>
</dashboard>
```

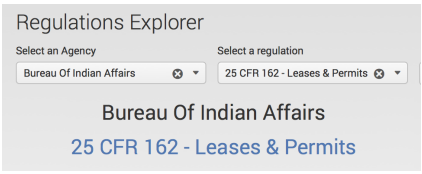
渡された値を受け付けるフォームのソースコード：

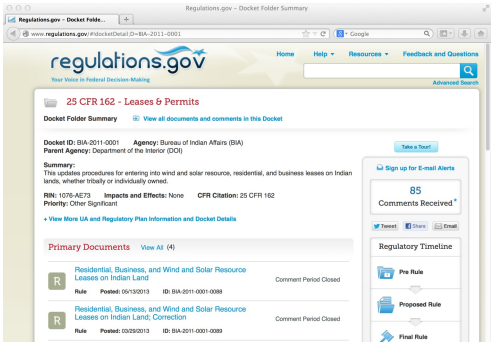
```
<form>
  <label>Dynamic Drilldown (Target Form)</label>
  <description/>
  <fieldset submitButton="false" autoRun="true">
    <input type="text" token="src_type_tok" searchWhenChanged="true">
      <label>Source type</label>
    </input>
  </fieldset>
  <row>
    <panel>
      <chart>
        <title>Source type details</title>
        <search>
          <query>
            index=_internal | timechart span=1week count by $src_type_tok$
```

```
        </query>
        <earliest>-30d@d</earliest>
        <latest>now</latest>
      </search>
      <option name="charting.chart">column</option>
    </chart>
  </panel>
 </row>
</form>
```

非表示フィールドを使った単一値ドリルダウン

単一値視覚エフェクトから、非表示のフィールドにドリルダウンすることができます。この例は、オンラインの政府規制文書にアクセスする App からのものです。単一値視覚エフェクトを使用して、選択した規制を表示します。規制をクリックすると、新しいブラウザウィンドウに政府の規制 Web サイトが開かれて、その規制のオンライン文書が表示されます。





例の App は、政府機関、規制、および規制 ID に関する情報を返すグローバルサーチを使用しています。後処理サーチを使って表示する値を取得する、2 つの単一値視覚エフェクトが含まれています。

2 種類のドロップダウンを利用できます。

- Select an agency
  政府機関を選択します。単一値視覚エフェクトとして選択された機関名を表示します。
- Select a regulation
  選択した機関の規制を選択します。規制名を単一値視覚エフェクトとして表示します。

2 番目の単一値視覚エフェクトは、その後処理サーチからフィールド `regulation_docketTitle` および `docketId` を使用します。ただし、単一値フィールドは最初に返された値しか表示できません (この例では `regulation_docketTitle`)。

視覚エフェクトは <drilldown> エレメントを使って、「隠された (非表示の) フィールド」`docketId` をドリルダウンします。`$row.<field>$` ドリルダウンイベントトークン内の隠されたフィールドを指定します。すべてのドリルダウンイベントトークンの一覧については、「単一イベントトークン」を参照してください。

$row.docketId$

以下のソースコードは、単一値視覚エフェクトの隠された値フィールドへのアクセス方法を表しています。

```
<form stylesheet="regulations_explorer.css">
  <label>Regulations Explorer</label>

  <fieldset autoRun="true" submitButton="false">
    <input type="dropdown" token="agency" searchWhenChanged="true">
      <label>Select an Agency</label>
      <search>
        <query><!-- populating search for input --></query>
        <earliest>$earliest$</earliest>
        <latest>$latest$</latest>
        <fieldForValue>agencyName</fieldForValue>
```

```
        <fieldForLabel>agencyName</fieldForLabel>
      </search>
      <choice value="*">ALL</choice>
      <default>*</default>
    </input>

    <input type="dropdown" token="docket" searchWhenChanged="true">
      <label>Select a regulation</label>
      <search>
        <!-- populating search for input -->
      </search>
      <fieldForValue>docketTitle</fieldForValue>
      <fieldForLabel>docketTitle</fieldForLabel>
    </input>

    <!-- time picker input -->
  </fieldset>


  <!-- Global search for post process                       -->
  <!-- Provides docketId and regulation_docketTitle fields  -->
  <!-- That are consumed by the single value visualization  -->
  <search id="baseSearch">
    <query>
    | pivot regulations Regulations_Data count(Regulations_Data)
    AS "Count of Regulations Data" SPLITROW docketId
    AS "docketId" SPLITROW docketTitle
    AS "regulation_docketTitle" SPLITROW commentStatus
    AS "regulation_comment_status" SPLITROW commentEndDateLong
    AS "regulation_comments_end_date" SPLITROW commentStartDateLong
    AS "regulation_comment_start_date" SPLITROW agency_name
    AS "agency_name" FILTER docketTitle contains $docket|s$
    | sort - regulation_comment_start_date| head 1
    </query>
  </search>
  <row>
    <panel>
      <single>
        <!-- Displays regulation_docket title -->
        <search base="baseSearch">
          <query>
            | fields regulation_docketTitle, docketId
          </query>
          <earliest>$earliest$</earliest>
          <latest>$latest$</latest>

        </search>

        <drilldown>
          <link>
            <![CDATA[ http://www.regulations.gov/#!docketDetail;D=]]>$row.docketId$
          </link>
        </drilldown>
      </single>
    </panel>
  </row>
</form>
```

状況に応じたドリルダウンエレメント

状況に応じたドリルダウンは、同じダッシュボード上の視覚エフェクトへの結果を生成します。あるダッシュボードから別個のフォームにドリルダウン結果を生成する、前述の動的ドリルダウンの例と比べてみてください。<condition> エレメントと <drilldown>、<set>、および <unset> エレメントを併用して、状況に応じたドリルダウンを実現します。

<condition> エレメントを <drilldown> エレメントの子として使用します。<condition> エレメントの `field` 属性には、値を取得するフィールドを指定します。<condition> エレメントには、クリックされたフィールドに応じて、異なるドリルダウンアクションを指定することができます。

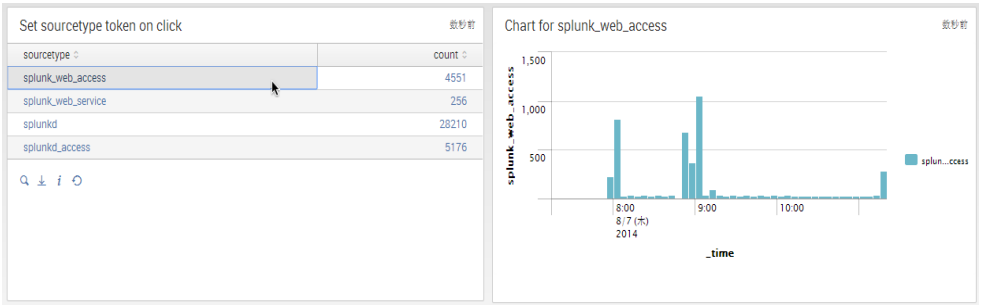
<set> トークンは、ドリルダウントークンからの値を、ドリルダウンのターゲットが使用する他のトークンに割り当てるために使用します。<set> エレメントは、<condition> エレメントの子です。<unset> エレメントは、前に設定されたトークンを削除します。

パネル視覚エフェクトの `depends` と `rejects` は、視覚エフェクトを表示するために必要なトークンの指定に使用し

ます。

状況に応じたドリルダウンの基本的な例

この例は、テーブル内の行の任意の場所がクリックされたら、同じページのグラフに値を渡す方法を表しています。ドリルダウンは句レックされた行の最初の列から値を取得して、グラフにそれを渡します。グラフは、ユーザーがテーブルをクリックするまでは非表示になっています。



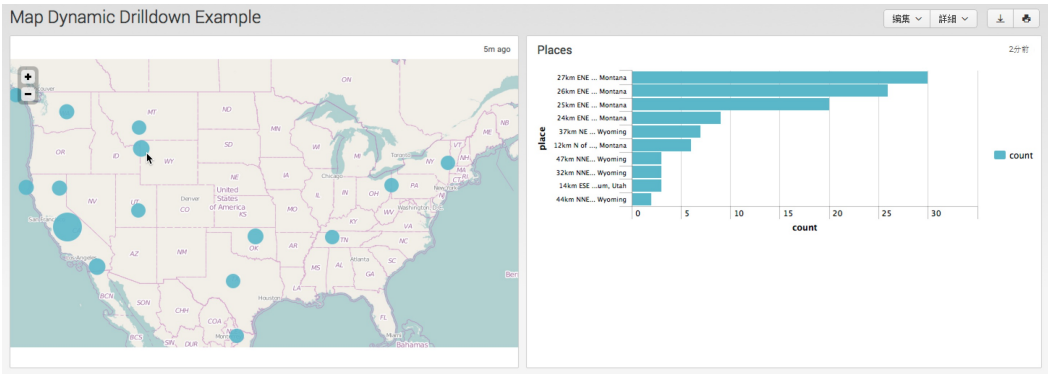
この例は、<set> エレメントを使って `src_type_tok` に `$click.value$` ドリルダウントークンから返された値 (テーブル内の最初の列の値) を設定します。「テーブルイベントトークン」を参照してください。

グラフはdepends 属性内の `src_type_tok` を、<chart> エレメント、<title> エレメント、およびサーチ内で使用します。depends 属性は、ユーザーがテーブル内をクリックするまで、グラフの表示を防止します。

```
<dashboard>
  <label>Contextual drilldown</label>
  <row>
    <panel>
      <table>
        <title>Set sourcetype token on click</title>
        <search>
          <query>
            index=_internal | stats count by sourcetype
          </query>
          <earliest>-4h</earliest>
          <latest>now</latest>
        </search>
        <drilldown>
            <set token="src_type_tok">$click.value$</set>
        </drilldown>
      </table>
    </panel>
    <panel>
      <chart depends="$src_type_tok$">
        <title>Chart for $src_type_tok$</title>
        <search>
          <query>
            index=_internal sourcetype=$src_type_tok$
            | timechart count by sourcetype
          </query>
          <earliest>-4h</earliest>
          <latest>now</latest>
        </search>
      </chart>
    </panel>
  </row>
</dashboard>
```

マップ視覚エフェクトからの状況依存の例

この例は、マップ視覚エフェクトでマーカーをドリルダウンする方法を表しています。マップ視覚エフェクトは、過去 1 ヶ月の地震アクティビティを表しています。マップマーカー上で生成されたサーチは、マップデータからの詳細情報を横棒グラフに表示します。たとえば、Montana、Utah、および Wyoming にまたがるマーカーをクリックすると、右側にグラフが表示されます。



注意：この例では、USGS Earthquakes Web サイトからダウンロードした地震データを使用しています。

以下のサーチは、マグニチュード .9 を超える地震を表示します。

```
index=main mag > .9 | geostats latfield=latitude longfield=longitude count
```

<drilldown> エレメントは、クラスタ化された場所を表すマーカーの境界に基づいて、トークンを設定します。取得される値は、`click.bounds.<orientation>` マップトークンから派生しています。ドリルダウンに利用できるすべてのマップトークンの詳細は、「マップイベントトークン」を参照してください。

```
<drilldown>
  <set token="bounds.north" > $click.bounds.north$</set>
  <set token="bounds.east"  > $click.bounds.east$</set>
  <set token="bounds.south" > $click.bounds.south$</set>
  <set token="bounds.west"  > $click.bounds.west$</set>
</drilldown>
```

グラフには以下のサーチが含まれています。このサーチは、ドリルダウンアクションが生成したトークンを使用します。

```
index=main mag > .9 | search latitude >= $bounds.south$ latitude < $bounds.north$ longitude >= $bounds.west$
longitude < $bounds.east$ | top place
```

この状況に応じたドリルダウン例に実装されているソースコードを以下に示します。

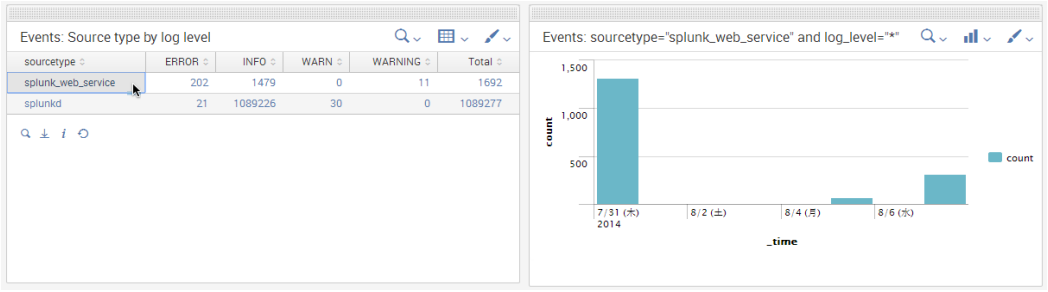
```
<row>
  <panel>
    <map>
      <search>
        <query>
        index=main mag>.9
        | geostats latfield=latitude longfield=longitude count
        </query>
        <earliest>0</earliest>
        <latest />
      </search>
      <option name="mapping.data.maxClusters">1000</option>
      <option name="mapping.drilldown">all</option>
      <option name="mapping.map.center">(39.3,-95.98)</option>
      <option name="mapping.map.zoom">4</option>
      <option name="mapping.markerLayer.markerMaxSize">40</option>
      <option name="mapping.markerLayer.markerMinSize">20</option>
      <option name="mapping.markerLayer.markerOpacity">0.9</option>
      <option name="mapping.tileLayer.maxZoom">7</option>
      <option name="mapping.tileLayer.minZoom">0</option>
      <drilldown>
        <set token="bounds.north">$click.bounds.north$</set>
        <set token="bounds.east">$click.bounds.east$</set>
        <set token="bounds.south">$click.bounds.south$</set>
        <set token="bounds.west">$click.bounds.west$</set>
      </drilldown>
      <option name="mapping.tileLayer.url">
        http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png
      </option>
    </map>
  </panel>
  <panel>
```

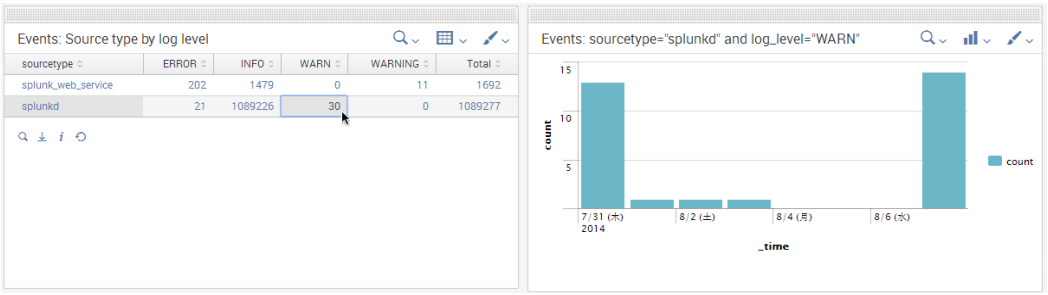
```
    <chart>
      <title>Places</title>
      <search>
        <query>
          index=main mag>.9 | search
          latitude >= $bounds.south$
          latitude &lt; $bounds.north$
          longitude >= $bounds.west$
          longitude &lt; $bounds.east$
          | top place
        </query>
        <earliest>0</earliest>
        <latest />
      </search>
      <option name="charting.chart">bar</option>
    </chart>
  </panel>
</row>
```

複数の条件を使った状況依存の例

この例は、ドリルダウンに複数の条件を設定します。ログレベルでの、ソースタイプのイベント数を記載したテーブルが含まれています。そのテーブル内をクリックすると、詳細なグラフが表示されます。詳細グラフは、ユーザーがテーブルをドリルダウンするまでは、表示されません。詳細グラフの内容は、ユーザーがテーブル内のどこをクリックしたのかによって異なります。

- [sourcetype] または [Total] 列をクリック
  詳細グラフには、すべてのログレベルの詳細情報が表示されます。
- [log level] 列をクリック
  詳細グラフには、そのログレベルの詳細情報が表示されます。





この例は、<condition> タグのフィールド属性を使って 3 つの条件を設定しています。各条件が、$s_sourcetype$ および s_log_level のトークン値を設定しています。詳細グラフのサーチは、これらのトークンを使用します。

```
  <drilldown>
    <condition field="sourcetype">
      <set token="s_sourcetype">$row.sourcetype$</set>
      <set token="s_log_level">*</set>
    </condition>
    <condition field="Total">
      <set token="s_sourcetype">$row.sourcetype$</set>
      <set token="s_log_level">*</set>
    </condition>
    <condition field="*">
      <set token="s_sourcetype">$row.sourcetype$</set>
      <set token="s_log_level">$click.name2$</set>
```

```
    </condition>
  </drilldown>
```

テーブル内のすべての列に対して、トークン $s_sourcetype$ は $row.sourcetype$ テーブルトークンからの値を取得します。これは、クリックされたセルのソースタイプを値に設定します。

[sourcetype] と [Total] 列の場合、クリックによりs_log_level トークンの値に「*」が設定されます。

[log level] 列の場合、クリックにより s_log_level トークンの値に $click.name2$ の値が設定されます。このトークンは、クリックされたテーブルセルの列の名前を取得します。

詳細グラフの <chart> エレメントは、`depends` 属性の値に $s_sourcetype$ を設定します。グラフは、テーブルのドリルダウンによりこのトークンが設定されるまで表示されません。

<chart depends="$s_sourcetype$">

この動的ドリルダウン例に実装されているソースコードを以下に示します。

```
<dashboard>
  <label>Contextual Example with Multiple Conditons</label>
  <row>
    <panel>
      <table>
        <title>Events: Source type by log level</title>
        <search>
          <query>
          index=_internal log_level=*
          | chart count over sourcetype by log_level | addtotals
          </query>
          <earliest>-7d@h</earliest>
          <latest>now</latest>
        </search>
        <option name="drilldown">cell</option>
        <drilldown>
          <condition field="sourcetype">
            <set token="s_sourcetype">$row.sourcetype$</set>
            <set token="s_log_level">*</set>
          </condition>
          <condition field="Total">
            <set token="s_sourcetype">$row.sourcetype$</set>
            <set token="s_log_level">*</set>
          </condition>
          <condition field="*">
            <set token="s_sourcetype">$row.sourcetype$</set>
            <set token="s_log_level">$click.name2$</set>
          </condition>
        </drilldown>
      </table>
    </panel>
    <panel>
      <chart depends="$s_sourcetype$">
        <title>
          Events: sourcetype="$s_sourcetype$" and log_level="$s_log_level$"
        </title>
        <search>
          <query>
          index=_internal sourcetype="$s_sourcetype$"
          log_level="$s_log_level$" | timechart count
          </query>
          <earliest>-7d@h</earliest>
          <latest>now</latest>
        </search>
      </chart>
    </panel>
  </row>
</dashboard>
```

グラフのコントロール

ここでは、グラフ内でデータを表示する際の高度な動作について説明しています。

パンとズームによるグラフコントロール

パンおよびズーム機能により、グラフの詳細を強調表示したり、必要に応じて詳細を別個のパネルに表示したりできます。パンおよびズーム機能は、以下のグラフで利用できます。

縦棒
折れ線
面

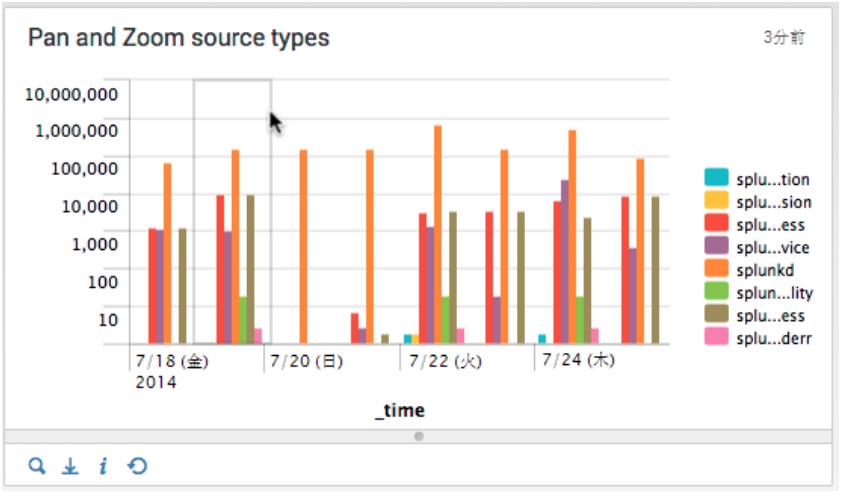
パンおよびズーム機能へのアクセス方法の例を以下に示します。

パンとズームの動作

以下のダッシュボードは、7 日間のソースタイプを記載するグラフを表示します。Y 軸は対数スケールを使って、より分かりやすい画像を提供しています。このパネルには以下のサーチを指定します。

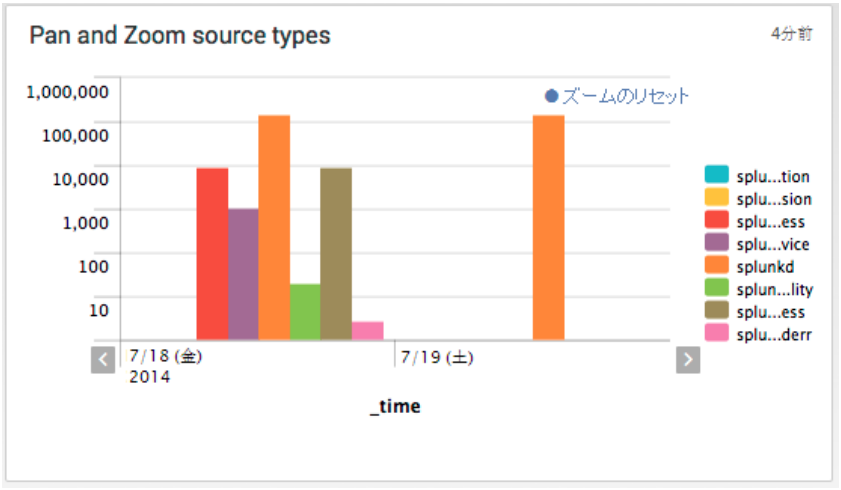
```
index=_internal | timechart count by sourcetype
```

以下の画面は、2 日間の結果選択を表しています。



結果となるグラフでは、選択項目にズームインして、選択した領域の詳細が表示されます。

- X 軸の左矢印と右矢印を使って、選択期間を前後に移動します。
- **[ズームのリセット]** をクリックすると、元のグラフに戻ります。

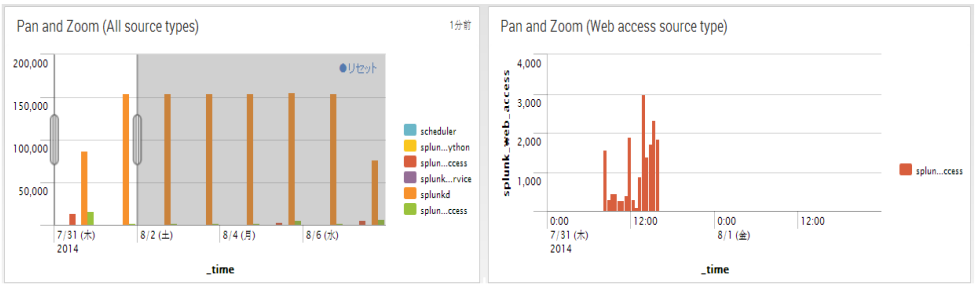


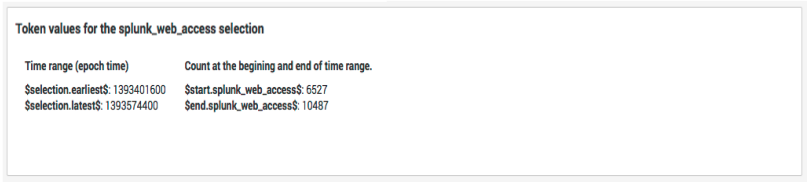
他のグラフへのズーム

別のグラフに結果を表示するように、パンとズーム機能を指定することができます。以下の例は、前述の「パンとズーム動作」と同じ基本例を使用しています。左側のグラフにはすべてのソースタイプが表示され、1 日間の期間が選択されています。他のグラフには、選択した期間の `splunk_web_access` ソースタイプのみが表示されます。

左側のグラフの時間範囲の端をドラッグして、期間を拡大することができます。また、選択した時間範囲を左右に移動して、前のまたは後の時間範囲を指示することもできます。

下のグラフには、パンとズーム動作を実装したトークンの値が表示されます。





実装の詳細

ズーム結果を別個のグラフに表示するには、まずベースとなるグラフのシンプル XML を編集します。選択した時間範囲のトークン値を設定するには、<selection> エレメントを使用します。

> **注意：**トークンの詳細は、「ダッシュボードでのトークンの使用」を参照してください。「パン/ズーム・コントロールのトークンの定義」では、パン/ズーム動作特有のトークンについて説明しています。

`$start$`
`$end$`
選択された時間範囲の開始/終了時の X 軸の値を取得する、事前定義されたトークン。この例では、時間グラフの開始/終了時刻を取得します。この値はエポック時です。

`$start.splunk_web_access$`
`$end.splunk_web_access$`
選択範囲の開始/終了時に、指定したシリーズの Y 軸の値を取得します。この例で、値は `splunk_web_access` フィールドのイベント数です。

`start` および `end` トークンは、グラフでのみ有効になります。ダッシュボード内の値にアクセスできるように、定義したトークンに値を割り当てます。

```
<chart>
  <title>Pan and Zoom (All source types)</title>
  <searchString>
    index=_internal | timechart count by sourcetype
  </searchString>
  <earliestTime>-7d@h</earliestTime>
  <latestTime>now</latestTime>
  . . .
  <selection>
    <set token="selection_earliest">$start$</set>
    <set token="selection_latest">$end$</set>
    <set token="start_splunk_web_access">$start.splunk_web_access$</set>
    <set token="end_splunk_web_access">$end.splunk_web_access$</set>
  </selection>
  . . .
</chart>
```

ターゲット・グラフで、選択時間範囲にアクセスするには $selection_earliest$ および $selection_latest$ を使用します。

```
<chart>
  <title>Pan and Zoom (Web access source type)</title>
  <search>
    <query>
      index=_internal sourcetype=splunk_web_access
      |  timechart count by sourcetype
    </query>
    <earliest>$selection_earliest$</earliest>
    <latest>$selection_latest$</latest>
  </search>
```

```
  . . .
</chart>
```

HTML パネルには、$start$ および $selection$ トークンが取得した値が表示されます。

```
<html>
  <h3>Token values for the splunk_web_access selection</h3>
  <table border="0" cellpadding="12" cellspacing="0">
  <tr>
    <td>
        <p><b>Time range (epoch time)</b></p>
        <p><b>$$selection_earliest$$</b>: $selection_earliest$
        <br /><b>$$selection_latest$$</b>: $selection_latest$</p>
    </td>
    <td>
        <p><b>Count at the begining and end of time range.</b></p>
        <p><b>$$start_splunk_web_access$$</b>: $start_splunk_web_access $
        <br /><b>$$end_splunk_web_access$$</b>: $end_splunk_web_access$</p>
    </td>
  </tr>
</table>
</html>
```

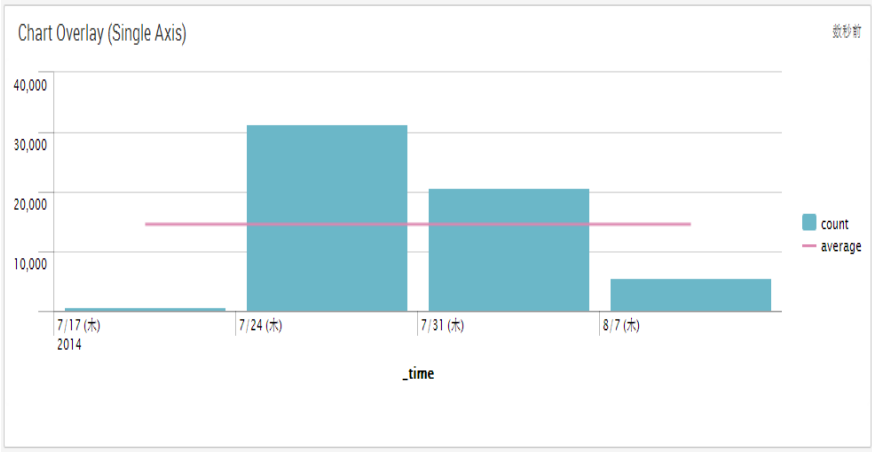
グラフのオーバーレイ

グラフのオーバーレイ機能を使って、単一のグラフに 2 つの異なるシリーズを表示することができます。縦棒グラフ、面グラフ、または別の折れ線グラフ上に、サーチ結果のシリーズを折れ線グラフで表示することができます。

オーバーレイを使用する場合、表示する値を 1 軸または 2 軸にすることができます。1 軸では、オーバーレイする値とサーチ結果を、同じ Y 軸に対して描画します。2 軸の場合、オーバーレイする値を表す 2 番目の Y 軸を指定します。

グラフのオーバーレイの例 (1 軸)

この例は、1ヶ月の時間グラフ上に `splunk_web_access` ソースタイプイベント数を週単位で表示します。このグラフにオーバーレイされるのは、これらのイベントの週単位の平均数です。



このグラフを作成するサーチは以下のようになります。

```
index=_internal sourcetype=splunk_web_access | timechart span=1week count | eventstats avg(count) as average | eval
average=round(average,0)
```

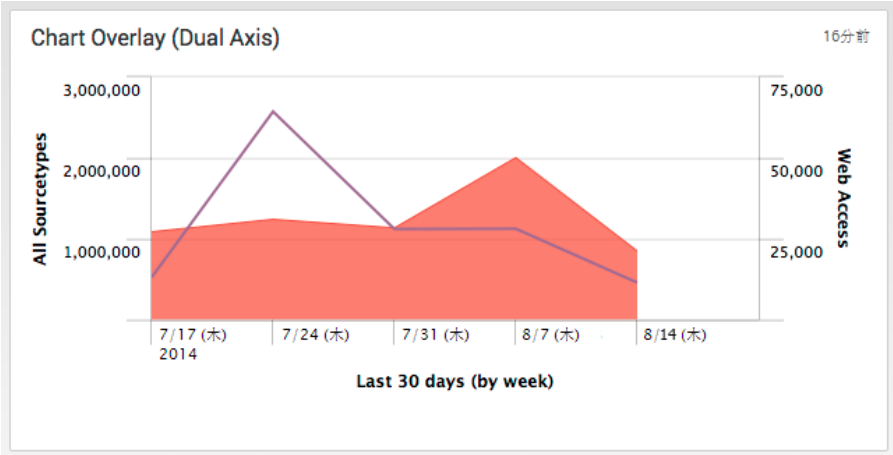
ビジュアルエディタを使ってオーバーレイを作成することができます。

1. ダッシュボードから、**[編集]** > **[パネルの編集]** をクリックします。
2. パネルを追加して、以下の項目を指定します。
    - **コンテンツのタイトル**：グラフのオーバーレイ (1 軸)
    - **サーチ文字列**：上記のサーチ文字列。
    - **時間範囲**：30 日間。
3. グラフのオーバーレイパネルで、**[プロパティの編集]** アイコンをクリックします。**[グラフのオーバーレイ]** をクリックします。
4. **[オーバーレイ]** フィールド内をクリックします。オーバーレイとして選択できる利用可能なフィールドから、**[average]** を選択します。
5. **[軸として表示]** では、**[オフ]** をクリックします。

この例では、2 番目の Y 軸は指定しません。
6. [適用] をクリックします。[完了] をクリックします。

グラフのオーバーレイの例 (2 軸)

この例では、すべてのソースタイプ合計に対して、splunk_web_access ソースタイプのイベント数をオーバーレイします。グラフには、個別の Y 軸に対する Web アクセス合計を描画します。



このグラフを作成するサーチは以下のようになります。

```
index=_internal sourcetype=* | timechart span=1week count as "All Sourcetypes"
count(eval(sourcetype="splunk_web_access")) as "Web Access"
```

ビジュアルエディタを使ってオーバーレイを作成することができます。

1. ダッシュボードから、[編集] > [パネルの編集] をクリックします。
2. パネルを追加して、以下の項目を指定します。
   - **コンテンツのタイトル**：グラフのオーバーレイ (2 軸)
   - **サーチ文字列**：上記のサーチ文字列。
   - **時間範囲**：30 日間。
3. グラフのオーバーレイパネルで、[プロパティの編集] アイコンをクリックします。[グラフのオーバーレイ] をクリックします。
4. [オーバーレイ] フィールド内をクリックします。オーバーレイとして選択できる利用可能なフィールドから、[Web Access] を選択します。
5. [軸として表示] では、[オン] をクリックして、2 番目の Y 軸を指定します。
6. [タイトル] では、[カスタム] をクリックします。2 番目の軸を指定するため、隣のテキスト・フィールドに「Web Access」と入力します。
7. [スケール] で [継承] をクリックして、最初の Y 軸からのスケール選択を継承します。
8. [適用] をクリックします。[完了] をクリックします。

グラフの表示上の問題

このトピックは、グラフ視覚エフェクトを使った表示上の問題について説明していきます。

非変換コマンドを使ったサーチ

次のような変換コマンドを含まない、または適切に使用していないサーチをベースにしたグラフを生成することはできません：

```
chart
timechart
stats
eval
```

変換コマンドの詳細は、「変換コマンドとサーチについて」を参照してください。

時間グラフ

timechart コマンドを使ってのみ、時間ベースのデータを描画できます。他の変換コマンドを使って時間ベースのシリーズを描画すると、グラフはタイムスタンプを文字列のシリーズとして取り扱います。

サーチ結果の切り捨て

Splunk Enterprise はシリーズ当たりに返される結果数を制限するために、抑制機能を採用しています。デフォルト値は、シリーズあたりの結果数を最初の 1000 件に制限しています。グラフがこの制限値に達すると、結果が切り捨てられたことを示すメッセージが表示されます。

シンプル XML コードのデフォルト値に優先する設定を行うには、charting.data.count プロパティを使用します。詳細は、『グラフ設定リファレンス』の「全般的なグラフのプロパティ」を参照してください。

描画できるポイント数の制限

Web ブラウザのパフォーマンスへの悪影響を防止するために、Splunk Enterprise のグラフライブラリには、個別のグラフに対して描画できるポイント数に制限があります。Web ブラウザによって、このデータの切り捨てが行われる制限値は異なります。

グラフに切り詰められたデータセットが表示される場合、グラフの下に結果が切り捨てられたことを示すメッセージが表示されます。

ブラウザタイプ別のデフォルトの切り捨て制限を以下の表に示します。

Web ブラウザ	最大描画ポイント数
Chrome	20000
Firefox	20000
Safari	20000
Internet Explorer 7 Internet Explorer 8	2000
Internet Explorer 9 Internet Explorer 10 Internet Explorer 11	20000

描画できるポイント数のデフォルトの制限値に優先する設定を行うには、2 種類の方法があります。

- すべてのブラウザに対して強制する制限を設定する。
- グラフ単位に制限を設定する。

すべてのブラウザに対して強制する制限の設定

web.conf 設定ファイルは、異なるブラウザに対して描画できる最大ポイント数を示しています。ブラウザあたりの設定に優先する設定を行うには、web.conf にすべてのブラウザに対する最大ポイント数を定義します。以下の設定をコメント行から解除して、すべてのブラウザに対する制限を定義します。

jschart_truncation_limit = 20000

グラフ単位の制限の設定

グラフのシンプル XML を編集して、特定のグラフに描画できる最大ポイント数を設定することができます。<chart> エレメントに対して、charting.chart.resultTruncationLimit プロパティを編集します。詳細は、『グラフ設定リファレンス』の「面、横棒、縦棒、折れ線グラフ、および散布図」を参照してください。

カテゴリ制限

カテゴリ別にデータを描画する場合、Splunk グラフライブラリにはグラフのラベル表示に影響する制限があります。この制限は水平軸 (X 軸) と垂直軸 (Y 軸) とでは異なっています。

X 軸の各ラベルには、最低 20 ピクセルが必要です。Y 軸には、最低 15 ピクセルが必要です。これだけのピクセルが利用できない場合、ラベルは表示されません。

X 軸にズームインすると、カテゴリ制限により表示されないラベルを参照することができます。詳細は、「パンとズームグラフコントロール」を参照してください。

ダッシュボード：概要

Splunk Web フレームワーク

Splunk Web フレームワークは、ダッシュボードとフォームを作成するためのさまざまなオプションを提供しています。ダッシュボードの開発方法は、表示するデータ、データ表示の複雑さ、および開発環境によって異なります。

シンプル XML

デフォルトで、Splunk はシンプル XML を使ってダッシュボードを作成します。シンプル XML のダッシュボードは、Splunk の対話型編集機能を使って作成、変更することができます。コードを記述する必要はありません。ただし、一部のダッシュボード機能は、シンプル XML ソースを記述する場合にのみ利用できます。

シンプル XML：

- さまざまなデータを視覚化した任意の数のパネルを持つ、ダッシュボードやフォームを作成します。
- Splunk Web には、シンプル XML ダッシュボードとパネルを作成、変更するための対話型編集機能が用意されています。
- Splunk には、ソースコードの編集に利用できる XML エディタが用意されています。

- 視覚エフェクト用のドリルダウン機能。
- ダッシュボードから各種形式の PDF 生成が可能。

詳細は、以下の項目を参照してください。

- Splunk のダッシュボードとフォーム
- ダッシュボードエディタについて
- シンプル XML の編集について

シンプル XML の機能拡張

Splunk 6 から、シンプル XML でダッシュボードのレイアウトのカスタマイズ、新しい視覚エフェクトの追加、およびすべての機能を保持しながらのダッシュボードの動作の変更が可能になりました。

シンプル XML 機能拡張の概要：

- SplunkJS スタックアクセス
- カスタムの CSS スタイルシートと JavaScript ファイルで、レイアウトとスタイルの柔軟性が向上。
- ダッシュボード内のパネル全体のエレメントを同様にスタイル設定するためのトークンを利用可能。
- データをモデル化する独自の視覚エフェクトを作成可能。

詳細は、以下の項目を参照してください。

- シンプル XML のカスタマイズ
- Splunk Developer Portal (開発者ポータル) からの SplunkJS スタック
- Splunk 6 Dashboard Examples (ダッシュボードの例) App (Splunk Apps で利用可能)

SplunkJS スタックにアクセスする HTML

シンプル XML ダッシュボードを、フォーム入力、テーブル、およびグラフなどの SplunkJS から利用できる機能にアクセスする HTML に変換することができます。これにより、HTML や JavaScript を使用する Web 開発者環境で、完全なレイアウトコントロールを実現することができます。

SplunkJS Stack の特徴

- 完全なレイアウトコントロール
- HTML や JavaScript を使用する Web 開発者環境

詳細は、以下の項目を参照してください。

- ダッシュボードの HTML への変換
- Splunk Developer Portal (開発者ポータル) からの SplunkJS スタック

Django Bindings

Django Bindings を使って、Django のサーバー側機能、コンポーネント、テンプレートを使用するカスタムダッシュボードを作成することができます。Django Bindings の高度な機能を利用するには、Django Web フレームワークの知識が必要です。これにより、HTML や JavaScript では利用できないサーバー側機能にアクセスしたり、Django タグで再利用可能コンポーネントを作成したりすることができます。

Django Bindings の特徴：

- Django テンプレートおよびテンプレートタグは、Splunk ビューとサーチ管理を作成する手軽な手段を提供しています。
- Django Bindings は、サーバー側開発のための高度な機能を提供しています。

詳細は、以下の項目を参照してください。

- Splunk Developer Portal (開発者ポータル) からの Django Bindings

アドバンスト XML と Splunk モジュールシステム

Splunk 6 では、アドバンスト XML と Splunk モジュールシステムで作成された、従来の Splunk アプリケーションとダッシュボードを引き続きサポートしています。アドバンスト XML は、高度な Splunk ダッシュボード/アプリケーションを作成するために利用できる、設定/再利用可能なサーバー側モジュールを提供しています。

Splunk は引き続き アドバンスト XML をサポートしていますが、Splunk Web フレームワークの新たなコンポーネントを利用することをお勧めしています。

詳細は、以下の項目を参照してください。

- アドバンスト XML について

ダッシュボードとフォーム

Splunk Enterprise App の各ページは**ビュー**になります。たとえば、サーチとレポート App のサーチタイムラインページは、その App 出荷時のデフォルトのビューです。独自の App を設計する場合、それに対応するビューを作成することができます。既存の App にビューを追加することもできます。

もっとも一般的なビューがダッシュボードです。各ダッシュボードには 1 つまたは複数のパネルが含まれており、それぞれのパネルにグラフ、テーブル、イベントリスト、地図などの視覚エフェクトを表示することができます。各ダッシュボードパネルは、視覚エフェクトに結果を提供するために、ベースサーチを使用します。一般的に

はダッシュボードのロード時に、サーチが結果を返します。

フォームは、サーチにユーザー入力 (ドロップダウン・リスト、ラジオ・ボタン、テキスト・ボックスなど) を提供するダッシュボードです。フォームには、ダッシュボードで利用できるパネルや視覚エフェクトと同じオプションが含まれています。

ダッシュボードとフォームでは、パネルのサーチから異なる情報を抽出して強調するために、サーチを変更 (サーチの後処理) することもできます。

ダッシュボードとフォームの構造

ダッシュボードとフォームは、シンプル XML で作成できる 2 種類のビューです。構造は同じですが、いくつかの小さな違いがあります。以下のシンプル XML エレメントがダッシュボードやフォームを構成しています。これらのエレメントの大半は省略することができます。シンプル XML の詳細は、「シンプル XML リファレンス」を参照してください。

エレメント	説明
トップ・レベルのエレメント	<dashboard> または <form>
タイトル	<label> (オプション)
説明	<description> (オプション)
グローバル・サーチ	グローバル・サーチは、後処理サーチで利用します。後処理サーチには制限があります。「後処理の制限事項」を参照してください。「<search> エレメント」を参照してください。 <search id="[identifier]">
フォーム入力 (フォームのみ)	<fieldset> <input> <text> <time> <checkbox> <dropdown> <multiselect> <radio> <search> (入力選択項目を設定)
行	各行に 1 つまたは複数のパネルが含まれます。 <row>
パネル	各パネルには、タイトル (省略可)、入力 (省略可)、および 1 つまたは複数の視覚エフェクトが含まれます。利用できるパネルのタイプについては、「ダッシュボードのパネル」を参照してください。 <panel>
視覚エフェクト	視覚エフェクトには、サーチから返されたデータが表示されます。 <chart> <event> <map> <single> <table>
サーチ	視覚エフェクトのサーチ。「<search> エレメント」を参照してください。 <search> <search id="[identifier]"> 後処理サーチのベース・サーチ。 <search base="[*id*]"> ベース・サーチを参照する後処理サーチ。 <search ref="[*report*] [app="[*app name*]"]> レポートからのサーチを参照します。App への参照は省略することができます。
オプション	視覚エフェクト固有のプロパティ。 <option name="[option name]">

ダッシュボードとフォームの違い

行のレイアウト、パネル、およびパネル内の視覚エフェクトは基本的に同じです。シンプル XML でのダッシュボードとフォームの主な違いを以下に示します。

- それぞれが、異なるトップレベルのエレメント <dashboard> および <form> を持っています。
- フォームには、タイムレンジ・ピッカー、ドロップダウン・リスト、ラジオ・ボタン、テキスト・ボックスなどのユーザー入力があります。
- ソースコード内のシンプル XML エレメントの順序が、わずかに異なっています。

シンプル XML については、「シンプル XML リファレンス」を参照してください。<dashboard> と <form> のエントリを比較してください。

ダッシュボードのパネル

一般的にパネルには、パネル内の視覚エフェクトの内容を生成するサーチが含まれています。パネルには、以下のエレメントをオプションで使用することができます。

- タイトル
- サーチ
  パネルに表示するデータを生成する 1 つまたは複数のサーチ。さまざまなソースをサーチすることができます。
  - パネル・エディタを使って作成、編集したインライン・サーチ。
  - 埋め込みサーチまたはピボットを含むレポート。
- サーチ結果を変更するユーザー入力。
- データをグラフ、テーブル、またはチャートとして表示する視覚エフェクト。
- ユーザーにメッセージを表示する、HTML でエンコードされたテキスト。

インライン・パネル

インライン・パネルは、ダッシュボード・エディタやパネル・エディタで編集できるパネルです。シンプル XML のソース・コードを編集して、パネル・エレメントの子エレメントを編集することもできます。

ダッシュボード・エディタでインライン・パネルを作成することができます。また、サーチ、レポート、およびピボットから、ダッシュボードにインライン・パネルを追加することもできます。詳細は、「ダッシュボードへのパネルの追加」を参照してください。

プレビルト・パネル

プレビルト・パネルは、複数のダッシュボードで共有されるパネルです。各ダッシュボードは、パネルを表示するためのプレビルト・パネルへの参照を提供しています。

パネルをプレビルト・パネルに変換できます。シンプル XML コードを使って、共有可能なパネルを作成することもできます。

ダッシュボードは、<panel> エレメントの参照属性を使って、プレビルト・パネルを表示します。プレビルト・パネルが現在の App に所属していない場合は、オプションの <app> 属性を使用します。

```
<panel ref="SharedDataPanel" app="exampleApp" />
```

ダッシュボード・エディタでプレビルト・パネルを追加するには、利用可能なパネルのリストから選択します。パネル・エディタを使ってプレビルト・パネルを編集することはできません。

「参照によるパネルの作成と追加」を参照してください。

ダッシュボードへのパネルの追加

さまざまな方法でダッシュボードにパネルを追加することができます。

- ダッシュボード・エディタでは、追加するパネルのタイプを選択することができます。以下の事項を選択できます。
  - インライン・パネルを作成する。
  - パネルに参照を追加する。
  - レポートからパネルを追加する。
  - 他のダッシュボードからパネルを複製する。
- [サーチ] ページで、サーチ結果をインライン・パネルとして保存します。
  『サーチ・マニュアル』の「結果の保存」を参照してください。
- レポートの詳細ページで、レポートをインライン・パネルとして保存します。
  「レポートの作成と編集」を参照してください。
- ピボット・エディタで、ピボットをインライン・パネルとして保存します。

ビューについて

シンプル XML では、ビューをダッシュボードまたはフォームとして定義できます。ただし、Splunk Web Framework には、その他のタイプのビューもあります。

- **アドバンスト XML ビュー**
  従来のアドバンスト XML で作成されたダッシュボードは、インポートされた Mako テンプレートに基づいてビューを定義しています。ダッシュボードとフォーム以外の、カスタム Mako テンプレートからのビューなどの、他のビューを利用することもできます。詳細は、アドバンスト XML のレイアウト (Layout) テンプレートを参照してください。

- **HTML + SplunkJS Stack**
  シンプル XML ビューを、SplunkJS スタックにアクセスする HTML に変換することができます。変換後の HTML には、シンプル XML で定義されているダッシュボードやフォームビューの概念は適用されません。

Splunk Enterprise ダッシュボードのエディタ

Splunk Enterprise には、ダッシュボードを作成、編集するためのさまざまなオプションおよび視覚エフェクトが用意されています。ここでは、利用できるオプションの概要について説明しています。ダッシュボードを作成、編集するための Splunk Enterprise ツールの使用方法については、「Splunk Web を使ったダッシュボードの作成と編集」を参照してください。

ダッシュボードエディタ

ダッシュボードの作成、ダッシュボードへのパネルの追加、ダッシュボードの編集、フォームの作成、およびダッシュボードの PDF の生成を行うには、**ダッシュボードエディタ**を使用します。

フォームを作成するには、まずダッシュボードを作成して、次にダッシュボードに入力を追加します。詳細は、「ダッシュボード・エディタによるフォームの作成と編集」を参照してください。

ダッシュボード・エディタには、ダッシュボードを編集するための一連のダイアログとメニューが含まれています。

エディタ	説明
パネルの追加	以下のオプションから選択して、ダッシュボードにパネルを追加します。 • 新しいインライン・パネルを作成する。 • レポートに基づいたパネルを追加する。 • プレビルト・パネルに参照を追加する。 • 他のダッシュボードからパネルを複製する。
入力の追加	ダッシュボードにフォーム入力と [送信] ボタンを追加します。
パネル・エディタ	インライン・パネルを編集するための一連のダイアログです。パネルエディタでは、パネルプロパティの編集、パネルのベースサーチの表示と編集、視覚エフェクトの変更と設定などの作業を行えます。
ビジュアル・エディタ	視覚エフェクトを設定するための一連のダイアログです。視覚エフェクトによって、利用できるダイアログは異なります。[サーチ] ページと [レポート] ページでも、類似の編集ダイアログを利用することができます。[サーチ] と [レポート] から、ダッシュボードで使用する視覚エフェクトを定義することができます。

サーチ、レポート、またはピボットからのダッシュボードへのアクセス

ダッシュボードの作成、またはサーチ、レポート、またはピボットの保存時にパネルを追加することができます。

[サーチ] ページからパネルを追加する場合、インライン・サーチでパネルを作成します。元のサーチに影響を与えずに、パネル内のサーチ文字列を変更することができます。

レポートまたはピボット・エディタからパネルを追加する場合、インライン・サーチを使って、または直接レポートにリンクすることでパネルを作成できます。インライン・サーチを使ってパネルを作成した場合、元のレポートやピボットに影響を与えることなく、パネル内のサーチ文字列を変更することができます。レポートに直接リンクした場合、レポートが変更されるとパネルの内容も変更されます。

ピボットエディタ

ピボットから、視覚エフェクトの作成/編集用ツールである**ピボットエディタ**にアクセスすることができます。ピボットエディタは、ピボット内の定義を視覚エフェクトのプロパティに対応させるために、ビジュアルエディタよりも多くの視覚エフェクト定義オプションを提供しています。詳細は、「ピボット・エディタを使ったピボット・グラフと視覚エフェクトの設計」を参照してください。

Splunk Enterprise ソースエディタ

各種機能を利用するために、シンプル XML のソースコードの編集が必要な場合もあります。Splunk Enterprise には、シンプル XML や HTML を編集するためのソースエディタが用意されています。

ダッシュボードのシンプル XML を編集する場合、以下のような作業を行えます。

- さまざまなダッシュボード・パネルの書式設定プロパティを設定する。「グラフ設定リファレンス」を使って、グラフやゲージの外観をカスタマイズする。
- 場所マーカーを表示した地図を作成する。

- 高度な動的ドリルダウン動作を設定する (クリックすると別のダッシュボードを表示するドリルダウン操作など)。
- 静的なテキスト、画像、および HTML 書式を表示する HTML パネルを作成する。
- グラフをオーバーレイするパネルを設定する。Splunk Enterprise のグラフ・ライブラリには、オーバーレイ目的の特殊なグラフ・タイプが用意されています。

シンプル XML を使って洗練されたダッシュボードを作成する方法については、このマニュアルの「シンプル XML によるダッシュボードの作成と編集」を参照してください。

シンプル XML を使ったフォームの作成方法については、このマニュアルの「シンプル XML でのフォームの作成と編集」を参照してください。

ソース・コード・エディタ

お気に入りのソース・コード・エディタを使って、ダッシュボードのソース・コードを編集することができます。シンプル XML または HTML ソースの編集だけでなく、ダッシュボードがアクセスする CSS や JavaScript も編集できます。

このシナリオでは、ホストサーバー上の Splunk インスタンスにアクセスする必要があります。詳細は、「シンプル XML の編集について」を参照してください。

ダッシュボード作成用ワークフロー

Splunk はデータの分析と視覚化を行うための強力なプラットフォームです。Splunk には、データを取得して、それを高機能なダッシュボードに表現する豊富なツールが用意されています。

ダッシュボードやフォームの設計、作成時には、以下のワークフローに従うことをお勧めします。

1. **コンテンツの追加**
   ダッシュボードのベースとなるサーチを作成します。
2. **ユーザーインターフェイスの設計**
   ダッシュボード、フォーム、パネルを作成、変更します。
3. **対話機能の追加**
   サーチデータをドリル ダウンします。
4. **ダッシュボードのカスタマイズ**
   ダッシュボードにカスタマイズした機能を追加します。

コンテンツの追加

Splunk のサーチは、ダッシュボード、フォーム、そしてそれに含まれているデータの視覚エフェクトの基盤となっています。Splunk に用意されているデータの収集/分析用ツールを正しく理解する必要があります。

- **ダッシュボードのベースとなるサーチの作成**
  ご自分のデータの重要な観点を探し出し、自分の目標を達成するために必要なサーチを作成してください。Splunk のサーチ言語を初めて使用する場合は、『サーチ・マニュアル』のサーチの説明を参照してください。『サーチ・リファレンス』には、サーチ・コマンド早見表、一般的なサーチ・コマンドのリスト、および Splunk の各サーチ・コマンドの詳細情報など、Splunk を使ったサーチに関するさまざまな情報が記載されています。

- **レポートとして保存したサーチ**
  サーチをレポートとして保存して、ダッシュボードからレポートを参照することで、ダッシュボードでサーチを利用することができます。詳細は、『レポート・マニュアル』を参照してください。『レポート・マニュアル』には、詳細を説明しているセクション「レポートの作成と編集」があります。

- **ピボットでサーチを生成**
  ピボット・ツールを使用して、サーチをピボットとして生成することができます。このピボットは、レポートやダッシュボードにエクスポートできます。ピボットはデータモデルを使用することで、データセットを特定し、その構造に基づいてテーブル、グラフ、および他の視覚エフェクトを設計することができます。詳細は、『インストール・マニュアル』を参照してください。

- **再利用するパネルの作成**
  複数のダッシュボードに役立つ情報を収集するパネルを作成することができます。プレビルト・パネルにより、同じパネルを何回も作成/更新する無駄を回避することができます。パネルの内容が変更された場合、そのパネルを参照するすべてのインスタンスが更新後の情報を受け取ります。プレビルト・パネルにより、技術的な知識がないユーザーでも複雑なパネルを利用できます。プレビルト・パネルも含めたすべてのパネル・タイプに関する情報については、「ダッシュボードのパネル」を参照してください。

詳細は、以下の項目を参照してください。

- サーチ・マニュアル
  サーチについて
  より良いサーチの作成

- サーチ・リファレンス
  サーチ・リファレンス
  サーチ・コマンド早見表
  すべてのサーチ・コマンド

- レポート・マニュアル
  レポート・マニュアル
  レポートの作成と編集

- ピボット・マニュアル
  ピボット・マニュアル
  ピボット・エディタを使ったピボット・テーブルの設計

- ダッシュボードと視覚エフェクト
  ダッシュボードとフォームの構造
  Splunk Web でのダッシュボードの作成
  シンプル XML を使ったダッシュボードの作成

ユーザーインターフェイスの設計

ダッシュボードの設計に利用できるさまざまな対話型編集ツールが用意されています。ダッシュボード・エディタで直接ダッシュボードを作成することができます。サーチ、レポート、またはピボットから、新しいまたは既存のダッシュボードにパネルを追加できます。既存のダッシュボードから、または再利用目的で作成されたパネルを追加できます。

ダッシュボードエディタでは、パネルをドラッグアンドドロップしてレイアウトを変更することができます。ビジュアルエディタにアクセスして、パネルのタイトルの変更、データの視覚エフェクトの設定、パネルのサーチの編集を行えます。

詳細は、以下の項目を参照してください。

- ダッシュボードへのサーチ、レポート、またはピボットの追加
- ダッシュボードエディタを使ったダッシュボードの編集
- 視覚エフェクトの編集
- Splunk のダッシュボードとフォーム

対話機能の追加

Splunk Enterprise の対話ツールを使って、有益な各種ダッシュボードを作成することができます。ただし、本リリースでフォームを作成するには、シンプル XML ソースコードを編集する必要があります。フォームを作成する際には、まず Splunk の対話ツールを使ってダッシュボードのサーチや視覚エフェクトを設計します。次にソースエディタを使って、ダッシュボードをフォームに変換します。

視覚エフェクトには、さまざまなドリルダウン機能が用意されています。デフォルトのドリルダウン操作は、ビジュアル・エディタで設定することができます。動的ドリルダウンを使って、他のSplunk ビューや外部 Web ページにリンクすることもできます。動的ドリルダウンでは、パラメータをフォームやサードパーティの Web ページに送信して、宛先コンテンツをさらに有益な内容に強化することができます。動的ドリルダウンを利用するには、ダッシュボードまたはフォームのソース・コードを編集します。

詳細は、以下の項目を参照してください。

- Splunk のダッシュボードとフォーム
- ダッシュボードのフォームへの変換
- ビジュアルエディタで利用できるプロパティ
- ダッシュボードとフォームの動的なドリルダウン
- 「シンプル XML リファレンス」のドリルダウン・エレメント

ダッシュボードのカスタマイズ

シンプル XML には、レイアウトの変更、新しい視覚エフェクトの追加、およびダッシュボードの動作のカスタマイズを行うために修正できる箇所がいろいろと存在しています。これらのカスタマイズによって、コンテンツを表示するための有益なビューを作成することができます。

カスタマイズのためには、カスタム・スタイルシート、JavaScript、およびシンプル XML コードをいろいろな組み合わせで使用します。

- **CSS スタイル**
  App 内の個別のダッシュボードにカスタムスタイルシートを追加します。

- **レイアウト**
  ダッシュボードやフォームのエレメントの再配置やパネルの非表示などの、簡単なレイアウトの変更を行えます。

- **トークン**
  ダッシュボードページ全体に、独自のトークンを設定します。

- **カスタム視覚エフェクト**
  独自の視覚エフェクトをダッシュボードパネルとして作成します。

- **テーブルのセル表示**
  テーブルのセル内の、独自のスタイルと動作を指定します。

詳細は、以下の項目を参照してください。

- シンプル XML のカスタマイズ
- Splunk Developer Portal (開発者ポータル) からの SplunkJS スタック
- Splunk 6 Dashboard Examples (ダッシュボードの例) App (Splunk Apps で利用可能)

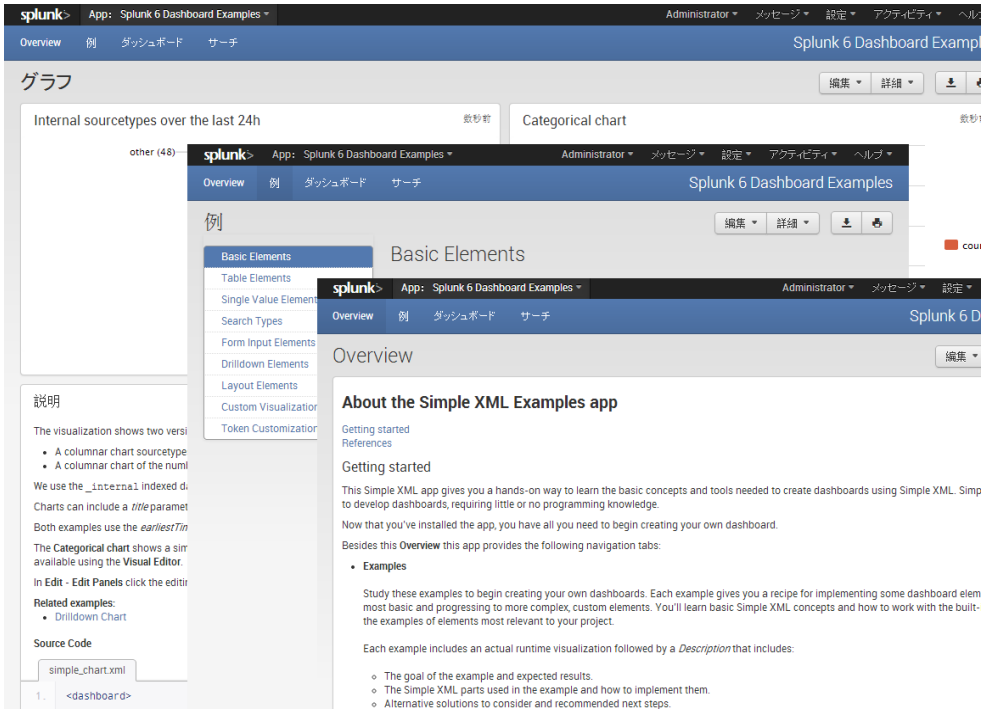
Dashboard Examples App

Splunk Appsで利用できる、Splunk 6 Dashboard Examples App は、シンプル XML でダッシュボードやフォームを作成するための、非常に優れたツールです。これには、基本的な機能や高度なカスタマイズ方法を理解

するための、さまざまな例が含まれています。それぞれの例には、ランタイム視覚エフェクト、例の説明、およびそのソースコードが含まれています。

Dashboard Examples App は、シンプル XML の初心者にも熟達した開発者にも役立ちます。

Splunk Apps から Splunk 6 Dashboard Examples App をダウンロードして、ご自分の Splunk インスタンスにインストールしてください。



Splunk SDK

Splunk Web フレームワークで作業を行う代わりに、Splunk にはご自分の開発環境で App を作成するための、豊富な SDK が用意されています。SDK は、Splunk Enterprise の REST API を使って Splunk Enterprise インスタンスにアクセスします。詳細は、「Splunk SDK の概要」を参照してください。

Splunk Enterprise 6.1 リリースで利用できる Splunk SDK：

- C#
- Java
- JavaScript
- PHP
- Python
- Ruby

Splunk Web でのダッシュボードの作成

ダッシュボードエディタについて

ダッシュボードエディタを使って、XML コードを記述しないでも、対話操作で**ダッシュボード**を作成、編集することができます。ダッシュボードエディタでは、以下の作業を行えます。

- ダッシュボードを作成する。
- パネルをダッシュボードに追加する。
- フォーム入力を追加してダッシュボードをフォームに変換する。
- ドラッグアンドドロップ操作でダッシュボードパネルの配置を変更する。
- ダッシュボードにデータを提供するサーチを編集する。
- パネルの各種視覚エフェクトを指定する。
- パネルの視覚エフェクトの書式設定オプションを指定する。
- ダッシュボードのソースコードを編集する。
- ダッシュボードを HTML に変換する。

ダッシュボードエディタの起動

新しいダッシュボードを作成するには：

1. App のダッシュボードページで、**[新しいダッシュボードの作成]** をクリックします。
   ダッシュボードは、App のコンテキストから作成します。ダッシュボードを作成したら、権限を編集して

ダッシュボードにアクセスできるユーザーを指定することができます。また、ダッシュボードを他の App コンテキストに移動することもできます。
2. **タイトル**、ID、および**説明**を指定します。権限を指定します。次に、**[ダッシュボードの作成]** をクリックします。
3. 続行するには、パネルの追加、入力の追加、またはダッシュボードのソース・コードの編集を行います。「ダッシュボードへのパネルの追加」および「ダッシュボード・エディタを使ったフォームの作成と編集」を参照してください。
4. **[完了]** をクリックして、ダッシュボードを作成します。

既存のダッシュボードを編集するには：

1. App のダッシュボードページから、既存のダッシュボードをクリックします。
2. **[編集]** をクリックして、ダッシュボード編集用のオプションを表示します。
ダッシュボードの編集の詳細は、「ダッシュボードエディタを使ったダッシュボードの編集」を参照してください。

ダッシュボードへのサーチ、レポート、またはピボットの追加

サーチ、レポート、またはピボットを、新しいダッシュボードまたは既存のダッシュボードのパネルとして追加することができます。

パネルは [サーチ]、[レポート]、または [ピボット] ページから直接追加することができます。

- [サーチ] ページまたは [ピボット] ページから、**[名前を付けて保存] > [ダッシュボードパネル]** をクリックします。
- [レポート] ページで、**[ダッシュボードに追加]** をクリックします。

ダッシュボードパネルの保存に利用できるオプションは、そのソース (サーチ、レポート、またはピボット) と、新しいダッシュボードを作成しているのか、または既存のダッシュボードにパネルを追加しているのかによって異なります。

> **警告：ダッシュボードとしてレポートにアクセスする場合の権限の変更**
> レポートを作成する場合、ご自分のユーザーコンテキストにより、レポートのサーチにアクセスする権限が決まります。次にダッシュボード内のパネルとしてレポートを保存すると、サーチにアクセスする権限が変更されます。レポートの作成者のみが利用できていた結果に、他のユーザーがアクセスできるようになります。そのため、ダッシュボード内で予期せぬデータ漏洩が発生する可能性があります。
>
> たとえば、admin ロールを持つユーザーは、`_internal` インデックスの内容を参照することができます。user ロールのユーザーは、このインデックスを参照する権限がありません。
>
> 管理者ユーザーが以下のサーチでレポートを作成した場合、レポートの結果は管理者ユーザーのみで共有され、アクセスすることができます。
> ```
> index=_internal | top sourcetype
> ```
> ただし、このレポートをダッシュボードとして保存すると、権限が共有され、user ロールを持つユーザーがインデックス `_internal` からの結果にアクセスできるようになります。レポートに対しては、このユーザーは `_internal` の内容を表示する権限がありません。
>
> また、role user を持つユーザーは、レポートにアクセスするためのダッシュボードを作成することで、`_internal` のコンテンツを参照することができます。

ダッシュボードパネルの視覚エフェクトの指定

新しいサーチの実行時またはレポートを開く際に、サーチの結果によって推奨できる視覚エフェクトは異なります。サーチに変換コマンドが含まれていない場合は、イベント・リストのみが利用できます。サーチを詳細モードで実行した場合は、レポート・サーチの場合でも、サーチのイベントの一覧を参照することができます。

サーチをダッシュボードパネルに追加する場合、パネルへの結果の表示方法を選択することができます。後ほどダッシュボードパネルエディタから、選択内容を変更することができます。

詳細は、「ダッシュボードエディタを使ったダッシュボードの編集」を参照してください。

ダッシュボードの権限の指定

ダッシュボードの作成時に、[ダッシュボードの作成] パネルからダッシュボードに対して以下の権限を指定することができます。

- **プライベート**
自分だけが、ダッシュボードを表示、編集する権限があります。
- **App 内で共有**
ダッシュボードは、それを作成した App の他のユーザーも利用できます。他のユーザーは、ダッシュボードを参照でき、権限によってはダッシュボードを編集することも可能です。

ダッシュボードの作成後は、権限を変更することができます。

1. [ダッシュボード] ページから、権限を編集するダッシュボードを探します。
2. [アクション] で、**[編集] > [権限の編集]** を選択します。
3. 以下の事項を指定します。
    - **所有者、App、またはすべての App に対して表示。**
    - **ユーザーの読み取りおよび書き込み権限**

割り当てられているユーザーロール (およびそのロールが保有する権限) によっては、定義できるアクセス権が制限されている場合もあります。

ダッシュボード権限の詳細

ダッシュボードは Splunk Enterprise のナレッジ・オブジェクトで、権限を設定、管理することができます。割り当てられているユーザーロール (およびそのロールが保有する権限) によっては、定義できるアクセス権が制限されている場合もあります。

たとえば、ユーザー・ロールがデフォルトの権限を持つ user の場合、自分のプライベートなダッシュボードしか作成できません。ただし、他のユーザーに読み取り/書き込みアクセスを与えることは可能です。

ユーザー・ロールがデフォルトの権限を持つ admin の場合、プライベートなダッシュボード、特定の App 内で利用できるダッシュボード、またはすべての App 内で利用できるダッシュボードを作成できます。他の Splunk ユーザーロールにアクセスさせることもできます。

ダッシュボードおよび他のナレッジ・オブジェクトの権限設定の詳細は、『ナレッジ管理』マニュアルの「ナレッジ・オブジェクトの権限の管理」を参照してください。

権限の編集例

管理 (admin) ユーザーによるダッシュボードの権限の設定方法の例を以下に示します。

注意： user など他のユーザーロールの場合、ダッシュボードエディタの権限の選択項目は、admin ユーザーが利用できる選択項目の一部分となります。

1. ダッシュボードで、**[編集]** を選択して、次に **[権限の編集]** を選択します。
2. ダッシュボードの権限を指定して、**[保存]** をクリックします。以下の項目から選択します。
   - **所有者**：ダッシュボードを作成したユーザーのみが、それを表示できます。
   - **App**：ダッシュボードは、それが作成された App 内でのみ参照することができます。ユーザーロールに対する読み取り/書き込み権限を指定します。
   - **すべての App**：すべての App でダッシュボードを表示できます。ユーザーロールに対する読み取り/書き込み権限を指定します。

ダッシュボードのフォームへの変換

フォームのシンプル XML はダッシュボードのシンプル XML と少し異なっています。ダッシュボードをフォームに変換するには、2 種類の方法があります。

- ダッシュボードにタイム・ピッカーまたはフォーム入力を追加する。
  シンプル XML が、ダッシュボードをフォームに変換するように更新されます。
- ダッシュボードにフォーム・エレメントを含めるように、ソースのシンプル XML を編集する。

ダッシュボードのカスタマイズ

ダッシュボードをカスタマイズして、ダッシュボードエディタでは利用できない機能を追加するには、さまざまな方法があります。

- 高度な機能を実装するように、シンプル XML を編集する。
  一般的に、対話型のエディタで利用できない視覚エフェクト機能を編集するには、シンプル XML を編集します。また、テキストの外観をカスタマイズするために、サーチ文字列からのトークンを利用することもできます。「シンプル XML の編集について」および「ダッシュボードでのトークンの使用」を参照してください。
- ダッシュボードのスタイルシートを編集する、または独自の CSS スタイルシートを追加する。
  「CSS、JavaScript、および他の静的ファイル」および「シンプル XML のカスタマイズ」を参照してください。
- ダッシュボード用に独自の JavaScript を追加します。
  「CSS、JavaScript、および他の静的ファイル」および「シンプル XML のカスタマイズ」を参照してください。
- ダッシュボードを HTML として変換またはエクスポートする。
  ダッシュボードの変換後、HTML コード、JavaScript、およびスタイルシートを編集して、カスタム動作を指定します。「ダッシュボードの HTML への変換」を参照してください。

ダッシュボードへのパネルの追加

ここでは、ダッシュボード・パネル、ダッシュボードに追加できるパネルのタイプ、およびパネルを別のタイプのパネルに変換する方法について説明していきます。

パネル、サーチ、ダッシュボード、およびフォーム

ダッシュボードには 1 つまたは複数のパネルが含まれています。ダッシュボードに入力を追加して、フォームを作成します。

一般的には、行に複数のパネルを配置します。サーチはパネルのコンテンツの基盤となっています。パネルには、サーチ結果がテーブルや視覚エフェクトで表示されます。

パネルの基盤となるサーチは、さまざまなソースから利用できます。

- パネルに対して指定されたインライン・サーチ。
- パネルに対して指定されたインライン・ピボット。
- レポートからのサーチ
  パネルには、サーチがあるレポートへの参照が含まれます。
- ピボットからのサーチ
  パネルには、ピボットがあるレポートへの参照が含まれます。
- プレビルト・パネルからのサーチ
  プレビルト・パネルはダッシュボードから参照します。

フォームまたはダッシュボードでは、すべてのパネルに適用されるグローバル・サーチを使用することができます。各パネル内では、グローバル・サーチを変更して結果を異なる方法で表示する、後処理サーチを使用します。後処理サーチには制限があります。「後処理サーチ」および「後処理の制限事項」を参照してください。

パネルのカテゴリ

パネルには 3 種類のカテゴリがあります。パネルのカテゴリに応じて、パネル・エディタを使ってパネルのサーチや視覚エフェクトを編集することができます。「ダッシュボードのパネルの編集」および「視覚エフェクトの編集」を参照してください。

インライン・パネル

インライン・パネルには、視覚エフェクトに表示するデータを生成する、1 つまたは複数のインライン・サーチが含まれています。サーチの作成と変更には、パネル・エディタを使用します。パネル・エディタで、データの視覚エフェクトを選択肢、視覚エフェクトのプロパティを設定します。

レポートからのパネル

レポートからのサーチと視覚エフェクトの両方に基づいてパネルを作成します。パネル内のサーチを変更することはできませんが、サーチ結果の視覚エフェクトを変更、設定することは可能です。レポート内のサーチが変更されると、そのレポートに基づくパネルにも変更が反映されます。レポートが使用しているリソースにアクセスできることを確認してください。

プレビルト・パネル

さまざまなダッシュボードで共有できるパネルを定義した、シンプル XML コード。ダッシュボードにプレビルト・パネルを表示するには、プレビルト・パネルへの参照を指定します。パネルのタイトル、サーチ、または視覚エフェクトを、ダッシュボードの参照から編集することはできません。

ダッシュボード・エディタを使ったパネルの追加

ダッシュボードにパネルを追加するには、ダッシュボードの **[編集]** メニューを使用します。**[編集]** メニューは、ダッシュボードのメニューから直接利用することも、[ダッシュボード] ページのダッシュボードのリストから利用することもできます。

1. ダッシュボードで、**[編集] > [パネルの編集]** を選択します。
2. **[パネルの追加]** を選択します。
3. いずれかのパネル・カテゴリを展開します。
   - **新規**
   - **レポートから新規作成**
   - **ダッシュボードから複製**
   - **プレビルト・パネルの追加**
4. (オプション) 特定のパネルをサーチするには、**[フィルタ]** にテキストを入力します。
   「フィルタリングによる利用可能なパネルの検索」を参照してください。
5. パネルを選択して、選択項目をプレビューします。
6. **[ダッシュボードに追加]** をクリックします。

フィルタリングによる利用可能なパネルの検索

サーチ・フィールドでフィルタを使用して、特定のパネルを検索、作成することができます。このサーチは、既存のダッシュボード、パネル、およびレポートから、特定の用語を持つものを検索します。指定したサーチ用語を使って検索したパネルの結果が表示され、その用語を含む既存のダッシュボードやパネルへのリンクが記載されます。

サーチのヒント：

- パネルのタイトルまたはパネル ID は、サーチに役立つ項目です。
- サーチをフィルタリングするには、視覚エフェクトのエレメント名、入力タイプ、およびその他のキーワードを使用してください。例：
  - マップ視覚エフェクトを利用しているダッシュボードを返す、またはマップ視覚エフェクトを利用した新しいパネルを作成する場合は、**map** でサーチします。
  - 複数選択フォーム入力を探すには、**multiselect** でサーチします。
- 複数の項目を使ってフィルタリングできますが、すべての項目がサーチ・フィールドに指定した順序で登場する必要があります。

ダッシュボードのパネルの再配置

ダッシュボード上のパネルを再配置するには、パネルをドラッグ・アンド・ドロップします。

1. ダッシュボードが編集モードになっていない場合は、**[編集] > [パネルの編集]** を選択します。
2. パネルを選択して、新しい位置にドラッグ・アンド・ドロップします。

ダッシュボードのインライン・パネルの作成

インライン・パネルを作成する場合、視覚エフェクトを選択して、パネルのサーチを指定します。

1. ダッシュボードで、**[編集] > [パネルの編集]** を選択します。
2. **[パネルの追加]** を選択します。
3. パネル・カテゴリ **[新規]** を展開して、データの視覚エフェクトを選択します。
4. (オプション) パネルのタイトルを入力します。
5. パネルに表示するデータを返すサーチ文字列を入力します。
    - (オプション) **[サーチの実行]** を選択して、サーチ結果をプレビューします。
6. サーチの時間範囲の選択
7. **[ダッシュボードに追加]** をクリックします。

レポートからのパネルの作成

レポートからパネルを作成する場合、利用可能なレポートのリストから選択します。

1. ダッシュボードで、**[編集] > [パネルの編集]** を選択します。
2. **[パネルの追加]** を選択します。
3. パネル・カテゴリ **[レポートから新規作成]** を展開して、利用可能なレポートを表示します。
   (オプション) **[フィルタ]** オプションを使って、特定のレポートをサーチします。「フィルタリングによる利用可能なパネルの検索」を参照してください。
4. プレビューするレポートを選択します。
5. **[ダッシュボードに追加]** をクリックします。

他のダッシュボードからのパネルの複製

他のダッシュボードからパネルを複製することができます。ダッシュボードに表示されるパネルは、複製したパネルと同じ編集権限を持っています。

1. ダッシュボードで、**[編集] > [パネルの編集]** を選択します。
2. **[パネルの追加]** を選択します。
3. パネル・カテゴリ **[ダッシュボードから複製]** を展開して、利用可能なレポートを表示します。
   (オプション) **[フィルタ]** オプションを使って、特定のパネルをサーチします。「フィルタリングによる利用可能なパネルの検索」を参照してください。
4. ダッシュボードを選択、展開します。プレビューするパネルを選択します。
5. **[ダッシュボードに追加]** をクリックします。

参照によるパネルの作成と追加

参照でダッシュボードに追加するパネルを作成することができます。このプレビルト・パネルは、パネルをさまざまなダッシュボードで再利用する場合に役立ちます。

他のダッシュボードから参照できるパネルを作成するには、2 種類の方法があります。

- 既存のパネルを、参照可能なプレビルト・パネルに変換する。
- [設定] ページで、シンプル XML コードを使ってパネルを作成する。

一般的には、ダッシュボード・エディタを使ってパネルを作成し、それをプレビルト・パネルに変換します。シンプル XML コードでパネルを作成することも可能です。

既存のパネルのプレビルト・パネルへの変換

パネルに後処理サーチが含まれていない場合にのみ、そのパネルをプレビルト・パネルに変換できます。後処理サーチは、他のサーチを参照するために base 属性を使っているサーチです。

1. 変換するパネルがあるダッシュボードで、**[編集] > [パネルの編集]** を選択します。
2. パネルの**オプション・メニュー**から、**[プレビルト・パネルに変換]** を選択します。
3. (オプション) 表示されたダイアログで、以下の項目を指定します。

    - ID：パネルのファイル名。英数字、「-」、および「_」のみを使用できます。
    - **パネルの権限**：**[プライベート]** または **[App 内で共有]** を選択します。
      プライベート：自分だけが、パネルを表示、編集する権限があります。
      App 内で共有：App の他のユーザーも、パネルを表示、編集することができます。

シンプル XML コードでのパネルの作成

シンプル XML コードを初めて使用する場合は、「シンプル XML の編集について」を参照してください。パネルの編集、設定方法の詳細は、『シンプル XML リファレンス』の「<panel>」、「パネル視覚エフェクト・エレメント」、および他の関連する項目を参照してください。

1. Splunk Web で、**[設定] > [ユーザー・インターフェイス] > [プレビルト・パネル]** を選択します。
2. [パネル] ページで、**[新規]** を選択して シンプル XML エディタを表示します。
3. シンプル XML エディタで、以下の事項を指定します。

    - **宛先 App**：パネルのコンテキストとなる App を選択します。

- **プレビルト・パネル ID**：パネルの名前を入力します。
  入力する名前は、ディスクにあるファイル名です。英数字、「-」、および「_」のみを使用できます。
- **プレビルト・パネル XML**：パネル・エレメントを定義するシンプル XML コード。
  参照パネル用のシンプル XML コードには、<panel> エレメントとその子エレメントのみを含めます。

ダッシュボードへのプレビルト・パネルの追加

1. ダッシュボードから、**[編集]** > **[パネルの編集]** を選択します。
2. **[パネルの追加]** を選択します。
3. パネル・カテゴリ **[プレビルト・パネルの追加]** を展開して、利用可能な参照パネルを表示します。
   (オプション) **[フィルタ]** オプションを使って、特定のパネルをサーチします。「フィルタリングによる利用可能なパネルの検索」を参照してください。
4. プレビューする参照パネルを選択します。
5. **[ダッシュボードに追加]** をクリックします。

プレビルト・パネルのインライン・パネルへの変換

プレビルト・パネルをインライン・パネルに変換することができます。プレビルト・パネルに、後処理サーチを含めることはできません。後処理サーチは、他のサーチを参照するために base 属性を使っているサーチです。

プレビルト・パネルをインライン・パネルに変換すると、サーチと視覚エフェクトをカスタマイズできます。

1. ダッシュボードから、**[編集]** > **[パネルの編集]** を選択します。
2. 変換するプレビルト・パネルの**オプション・メニュー**をクリックして、**[インライン・パネルに変換]** を選択します。

パネルのタイトルの編集

パネルには <title> エレメントがあります。これは、視覚エフェクトの <title> エレメントとは独立しています。パネルの作成時に、パネルのタイトルを指定できます。ダッシュボードに追加するパネルのタイトルは、1 つの例外を除いて編集できます。パネル・エディタを使ってプレビルト・パネルのタイトルを編集することはできません。

> **注意**：プレビルト・パネルを編集するには、パネルを **[設定]** > **[ユーザー・インターフェイス]** > **[プレビルト・パネル]** で開きます。

1. パネル・エディタで、パネルのタイトルをクリックします。
   パネルの **[オプション]** メニューを選択して、**[名前変更]** を選択することもできます。
2. パネルの新しい名前を入力します。

ダッシュボードからのパネルの削除

ダッシュボード・エディタを使って、またはシンプル XML コードを編集して、ダッシュボードからパネルを削除することができます。

- パネル編集モードのダッシュボード・エディタで、パネルの **[オプション]** メニューをクリックして、**[削除]** を選択します。
  または、パネルの右上にある**削除アイコン [X]** をクリックすることもできます。
- シンプル XML ソース・コードで、<panel> エレメントとそのコンテンツを削除します。

ダッシュボードエディタを使ったダッシュボードの編集

ここでは、ダッシュボード・エディタでの、ダッシュボードまたはフォームの基本的な編集操作について説明していきます。「視覚エフェクトの編集」では、ダッシュボードパネルの視覚エフェクトの作成、変更方法について説明しています。

「ダッシュボードのフォームへの変換」を除いて、ここのサブトピックはダッシュボードとフォームの両方に適用されます。

ダッシュボードのフォームへの変換

ダッシュボードのフォームへの変換手順を以下に示します。ダッシュボードに入力を追加すると、ダッシュボードがフォームに変換されます。

ダッシュボードからフォームを作成するには：

1. 1 つまたは複数のパネルを持つダッシュボードを作成します。
2. 編集モードになっていない場合は、**[編集]** > **[パネルの編集]** を選択します。
3. **[入力の追加]** メニューで、1 つまたは複数の入力を選択します。
4. 追加した各入力に対して、入力を編集して入力動作を指定します。
5. (オプション) 新しく作成するフォーム用に、入力をドラッグアンドドロップして再配置します。
6. (オプション) 入力をパネルにドラッグして、そのパネルにのみ入力を適用することを指定します。

フォームの作成と編集の詳細は、「ダッシュボード・エディタを使ったフォームの作成と編集」を参照してください。

ダッシュボードのパネルの編集

パネルを編集するために利用できるツールは、パネルの基盤となるベース・サーチによって異なります。パネル・

エディタには、各ベース・サーチのタイプに対応するアイコンが表示されます。

アイコン	サーチ
	インラインサーチ
	インラインピボット
	レポートからのサーチ
	レポートからのピボット

ダッシュボードパネルの基盤となるサーチを編集するには：

1. ダッシュボードが編集モードになっていない場合は、**[編集] > [パネルの編集]** を選択します。
   各パネルには、パネルのコンテンツを変更するための 3 つの編集アイコンが表示されます。
2. [パネルのプロパティ] アイコンを選択します。
   利用できるオプションは、ベースサーチの種類によって異なります。

   すべてのパネルタイプ

   - パネルのタイトルを編集します。
   - パネルを削除します。

   レポート

   - レポートを表示します。
   - [サーチ] または [ピボット] でサーチを表示します。
   - インラインサーチまたはピボットに複製します。
   - パネルに対して別のレポートを選択します。
   - このパネルのレポートに指定する視覚エフェクトを選択します。

   インラインサーチとインラインピボット

   - インラインサーチまたはインラインピボットを指定するサーチを編集します。
   - インラインサーチ/ピボットをレポートに変換します。

   サーチ、レポート、ピボットの詳細は、以下の資料を参照してください。

   - サーチマニュアル
   - サーチリファレンス：
   - レポートマニュアル
   - ピボットマニュアル
   - ピボットサーチコマンド

パネルの視覚エフェクトを変更するには：

1. ダッシュボードが編集モードになっていない場合は、**[編集] > [パネルの編集]** を選択します。
2. 視覚エフェクトアイコンをクリックして、視覚エフェクトを選択します。視覚エフェクト・アイコンの画像は、現在選択されている視覚エフェクトのタイプを表しています。

   Splunk は、利用可能な視覚エフェクト、およびサーチに対して推奨するエフェクトを表示します。

   サーチがピボットまたはピボット・レポートの場合、パネルの視覚エフェクトを変更することはできません。代わりにピボットエディタを使って視覚エフェクトを変更してください。詳細は、「ピボットエディタを使ったピボットグラフと視覚エフェクトの設計」を参照してください。

パネルの視覚エフェクトを設定するには：

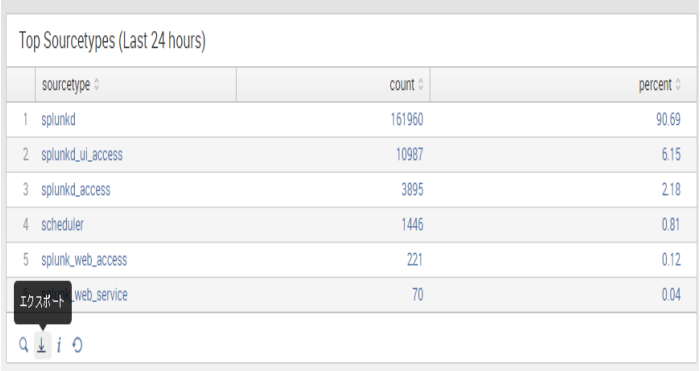
1. ダッシュボードが編集モードになっていない場合は、**[編集] > [パネルの編集]** を選択します。
2. [視覚エフェクトの書式設定] アイコンをクリックします。
   設定に利用できるプロパティは、視覚エフェクトによって異なります。
3. 視覚エフェクトを設定します。
   視覚エフェクトの書式設定の詳細は、「ビジュアルエディタで利用できるプロパティ」を参照してください。

視覚エフェクトのサーチの表示、エクスポート、調査

パネル・エディタでは、パネル内のデータを生成するサーチの詳細を参照することができます。以下のような作業を行えます。

- [サーチ] App でサーチを開く。
- ピボット・エディタでピボットを開く。
- サーチ結果をさまざまな形式でエクスポートする。
- サーチ・ジョブ調査でサーチを表示する。
- サーチを更新する。

これらの機能は、パネル上にマウス・カーソルを移動すると表示されるアイコンから利用できます。



視覚エフェクトのシンプル XML を編集して、これらの機能を無効にすることができます。「視覚エフェクトのサーチ・アクセス機能を無効にする」を参照してください。

[サーチ] App 内の視覚エフェクトのサーチの表示

[サーチ] App 内の視覚エフェクトのサーチを表示することができます。この機能は、サーチの詳細を調査したり、パネルのサーチ更新前に変更内容をテストしたりする場合に役立ちます。

視覚エフェクトがサーチではなくピボットを使用している場合は、ピボット・エディタでピボットを開きます。この手順は、サーチに基づく視覚エフェクトを例にしています。

視覚エフェクトのサーチまたはピボットを表示するには：

- パネル・エディタで、**[サーチで開く]** アイコンをクリックします。
  サーチを実行している [サーチ] App が、新しいウィンドウに表示されます。

サーチ結果の各種形式でのエクスポート

サーチの結果、または結果の一部をファイルに保存することができます。

パネルのサーチ結果をエクスポートするには：

1. パネル・エディタで、**[エクスポート]** アイコンをクリックします。
2. [結果のエクスポート] ダイアログで、以下の項目を指定します。
   - **フォーマット**：CVS、JSON、XML、または PDF。
     PDF は、レポートからのサーチでのみ利用できます。
   - **ファイル名**：(オプション) 結果を保存するファイル名。
   - **結果数**：**[制限]** または **[無制限]** を選択します。
3. **[エクスポート]** をクリックして、ローカル・ファイル・システムに結果を保存します。

サーチ・ジョブ調査でサーチを開く

サーチの詳細を表示するには、サーチ・ジョブ調査を使用します。「サーチ・ジョブ調査によるサーチ・ジョブ・プロパティの表示」を参照してください。

視覚エフェクトのサーチの詳細を表示するには：

- パネル・エディタで、**[調査]** アイコンをクリックします。
  サーチ・ジョブ調査が新しいウィンドウに表示されます。

サーチの更新

パネル内のサーチ結果を更新することができます。この機能は、最新の結果が表示されているかどうかを確認する場合に役立ちます。

パネルのサーチを更新するには：

- パネル・エディタで、**[更新]** アイコンをクリックします。

視覚エフェクトのサーチ・アクセス機能を無効にする

視覚エフェクトのシンプル XML コードを編集して、そのサーチ・アクセス機能を無効にすることができます。各視覚エフェクトには、このような機能を有効にする以下のプロパティが含まれています。『シンプル XML リファレンス』の、該当するパネル視覚エフェクト・エレメントを参照してください。

プロパティ	タイプ	デフォルト	説明
link.exportResults.visible	論理値	(説明を参照)	パネルの下部に [エクスポート] ボタンを表示します。 **デフォルト値**：link.visible の値。

link.inspectSearch.visible	論理値	(説明を参照)	パネルの下部に [調査] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.openPivot.visible	論理値	(説明を参照)	パネルの下部に [ピボットで開く] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.openSearch.search	サーチ文字列	—	[サーチで開く] ボタンで使用する代替サーチ。
link.openSearch.visible	論理値	(説明を参照)	パネルの下部に [サーチで開く] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.visible	論理値	true	パネルの下部にリンクボタンを表示します。

以下のコード・スニペットは、`link.visible` プロパティに `false` を設定して、グラフ視覚エフェクトのパネルの下部にある、すべてのサーチ・アクセス・アイコンを無効にします。

```
<panel>
   <chart>
     <title>Top sourcetypes in the last 24 hours</title>
     <search>
       <query>
         index=_internal group=per_sourcetype_thruput
         | chart sum(kb) by series
         | sort -sum(kb)</query>
       <earliest>-1d</earliest>
       <latest>now</latest>
     </search>
     <option name="charting.axisY.scale">log</option>
     <option name="link.visible">false</option>
   </chart>
 </panel>
```

ダッシュボードの編集操作

ダッシュボード全体に適用できる、さまざまな編集操作が用意されています。

ダッシュボードのソースコードの編集

ダッシュボードおよびそのパネルのソースコードを編集することもできます。シンプル XML をベースにしたダッシュボードの場合、ソースコードを編集して、ダッシュボードエディタでは利用できない機能にアクセスすることができます。HTML をベースにしたダッシュボードの場合、これはサードパーティ製のエディタを使わずにダッシュボードを編集できる唯一の方法です。

ダッシュボードのソースコードを編集するには：

1. ダッシュボードで、**[編集]** > **[ソースの編集]** を選択します。
   編集オプションの記載時に、ダッシュボードのタイプが表示されます。

シンプル XML ソースコードの編集については、「シンプル XML を使ったダッシュボードの作成」を参照してください。

HTML に変換またはエクスポートされたダッシュボードの編集については、「ダッシュボードの HTML への変換」を参照してください。

シンプル XML ダッシュボードの HTML への変換

デフォルトで、Splunk はシンプル XML をベースにしたダッシュボードを作成します。SplunkJS スタックにアクセスできるように、ダッシュボードを HTML ベースに変換/エクスポートすることができます。

> **注意：** Splunk のビジュアルエディタを使って、HTML ダッシュボードを編集することはできません。この HTML ダッシュボードに対して、統合 PDF 生成を利用することはできません。

ダッシュボードのソースコードを HTML に変換するには：

1. ダッシュボードで、**[編集]** > **[HTML に変換]** を選択します。
2. 以下の事項を指定して、**[ダッシュボードの変換]** をクリックします。

   - **[新規作成]** または **[現状と置換]**を選択します。
     現在のダッシュボードと置換すると、その基盤となるシンプル XML は利用できなくなります。新しいダッシュボードを作成して、元のシンプル XML コードは残しておくことをお勧めします。

- タイトル
- ID
- 説明
- 権限
  ダッシュボードを変換したら、その権限を編集することができます。

シンプル XML ソースコードの変換およびエクスポートした HTML の編集については、「ダッシュボードの HTML への変換」を参照してください。

ダッシュボードの複製

ダッシュボードを複製して、既存のダッシュボードのコピーを作成することができます。ダッシュボードの複製は、ダッシュボードの [編集] メニューで行えます。[編集] メニューは、ダッシュボードメニューから直接利用することも、[ダッシュボード] ページのダッシュボードのリストから利用することもできます。

ダッシュボードを複製するには：

1. 目的のダッシュボードで、**[編集]** > **[複製]** を選択します。
2. 新しいタイトル、ID、および説明を指定します。**[ダッシュボードの複製]** をクリックします。
3. ダッシュボードを複製したら、以下のような作業を行えます。

   - ダッシュボードの権限を表示、設定する。
   - ダッシュボードの PDF 配信をスケジュールする。
   - **[パネルの編集]** で、ダッシュボードを編集モードで開く。
   - **[表示]** で、ダッシュボードのコピーを表示する。

ダッシュボードの削除

ダッシュボードの削除は、ダッシュボードの [編集] メニューで行えます。[編集] メニューは、ダッシュボードメニューから直接利用することも、[ダッシュボード] ページのダッシュボードのリストから利用することもできます。

ダッシュボードを削除するには：

1. 目的のダッシュボードで、**[編集]** > **[削除]** を選択します。
2. ダッシュボードを削除することを確認して、**[削除]** をクリックします。

ダッシュボードの PDF 配信のスケジュール

ダッシュボードの PDF 配信のスケジュールは、ダッシュボードの [編集] メニューから行えます。[編集] メニューは、ダッシュボードメニューから直接利用することも、[ダッシュボード] ページのダッシュボードのリストから利用することもできます。

> **注意**：PDF 配信機能は、シンプル XML をベースにしたダッシュボードでのみ利用できます。HTML に変換/エクスポートしたダッシュボードの PDF 配信をスケジュールすることはできません。

ダッシュボードの PDF 配信をスケジュールするには：

1. 目的のダッシュボードで、**[編集]** > **[PDF 配信のスケジュール]** を選択します。
2. **[PDF 配信のスケジュール]** チェックボックスを選択/選択解除することによって、PDF 配信を有効/無効にすることができます。
3. PDF 配信を有効にした場合、以下のオプションを指定することができます。

   スケジュール
   配信先メールアドレス
   メール配信の件名 (`$name$` はダッシュボードのタイトルを示しています)
   用紙サイズ
   用紙レイアウト
4. (オプション) 配信をスケジュールする前に、テスト用メールを送信したり、サンプルの PDF を表示したりすることができます。
5. PDF 配信設定を保存するには、**[保存]** をクリックします。

詳細は、「ダッシュボード PDF の生成」を参照してください。

App レベルでのダッシュボード編集操作

App コンテキストに関する、さまざまなダッシュボード編集操作が存在しています。

ダッシュボードの App コンテキストの変更

ダッシュボードは、App のコンテキスト内に作成します。ダッシュボードにグローバル権限を設定しない限り、それを他の App から利用することはできません。ただし、ダッシュボードの App コンテキストを変更することは可能です。基本的にこれは、ダッシュボードをある App から他の App に移動することです。

> **注意**：この作業を行うことができるかどうかは、自分が保有するロールや権限によって決まります。

ダッシュボードの App コンテキストを変更するには：

1. Splunk Web で、**[設定]** > **[ユーザーインターフェイス]** > **[ビュー]** を選択します。
2. 移動するダッシュボードを探して、[アクション] の **[移動]** を選択します。
3. App コンテキストを選択します。**[移動]** をクリックします。

App のナビゲーションの指定

ダッシュボードの App のナビゲーションバーへの追加、一連のダッシュボードのナビゲーションバーのドロップダウンリストへのグループ化、App のデフォルトビューの指定を行うことができます。この作業を行うには、Splunk の [設定] メニューから、ナビゲーションメニューの XML を編集します。この機能を利用するには、適切なユーザーロールと権限が必要です。

> **注意：**ナビゲーションは、App 単位で管理されます。システム内のすべての App でグローバルに利用できるようにダッシュボードを昇格した場合、当初はそれらの App のトップレベルナビゲーションメニューの、デフォルトのドロップダウンリスト「未分類」のビューに表示されます。それらの App への書き込み権限を持つユーザーは、必要に応じて App のナビゲーションバー内の適切な位置にダッシュボードを移動することができます。

App のナビゲーションメニュー XML にアクセスするには：

1. 任意の Splunk ビューで、**[設定] > [ユーザーインターフェイス] > [ナビゲーションメニュー]** を選択します。
2. **[App コンテキスト]** から App を選択します。
3. 必要に応じて、**[所有者]** からユーザーロールを選択します。
   ユーザーロールは、App の権限設定によって異なります。
4. **[ナビゲーション名]** で **[デフォルト]** を選択して、ナビゲーションメニュー XML を Splunk ソースエディタに表示します。
   後述するように、ナビゲーションメニュー XML を編集します。

ナビゲーションバーへのダッシュボードの追加

ナビゲーションバーにダッシュボードを追加するには、<nav> エレメントの子として <view> エレメントにダッシュボードを指定します。App のデフォルトビューのナビゲーションバーにダッシュボード (`my_dashboard`) を追加する方法を以下に示します。

```
<nav search_view="search" color="#993300">
  <view name="search" default='true' />
  <view name="data_models" />
  <view name="reports" />
  <view name="alerts" />
  <view name="dashboards" />
  <view name="my_dashboard"/>
</nav>
```

<view> エレメントの `name` 属性には、ダッシュボードの ID を指定します。ダッシュボードの作成時には、この ID を指定します。ダッシュボード ID は、ダッシュボードにアクセスする URL の葉ノードを指定します。

ナビゲーションバーにダッシュボードのドロップダウンリストを作成するには、<collection> エレメントを使用します。

```
<view name="search" default="true"/>
 <view name="data_models" />
 <view name="reports" />
 <view name="alerts" />
 <view name="dashboards"/>
 <collection label="Shadow Dashboards">
   <view name="my_dashboard"/>
   <view name="my_other_dashboard"/>
 </collection>

</nav>
```

App のデフォルトビューの指定

App のデフォルトビューを定義するには、<view> エレメントに `default` 属性を指定します。ここに指定されたビューが、App のホームビューになります。デフォルトでは、[サーチ] ビューが App のホームビューになります。

たとえば、[Reports] ビューをホームビューに指定するには、以下のように指定します。

```
  <view name="reports" default="true"/>
```

ドリルダウン用デフォルトビューを指定するには：<nav> エレメントに `search_view` 属性を指定して、App のドリルダウン動作のデフォルトビューを定義します。これが、App のホームビューになります。ドリルダウン動作では、App 内のダッシュボードが特定のビューにリンクします (パネルのドリルダウン動作をカスタマイズした場合を除く)。

注意：一般的には、ドリル ダウン動作で [サーチ] ビューが表示されます。このデフォルト動作を変更する場合は、適切な注意を払うようにしてください。

App のナビゲーションバーの色の指定

App のナビゲーションバーの色を指定するには、<nav> エレメントの `color` 属性を使用します。例：

```
<nav search_view="search" color="#993300">
```

ダッシュボードエディタを使ったフォームの作成と編集

ダッシュボードに 1 つまたは複数の入力を追加して、フォームを作成します。フォーム作成の基本的なワークフローを以下に示します。実際の作業順序は、状況によって異なります。

1. ダッシュボードを作成して、1 つまたは複数の入力を追加します。
   「フォームを作成するための入力の追加」を参照してください。
2. (オプション) ラベルおよびトークンフィールドを編集します。
   フォーム内のサーチは、入力に定義されているトークンを使って、サーチから取得する値を指定します。
   「フォーム入力のトークン」を参照してください。
3. サーチ文字列内および他の場所にあるトークンを参照します。
   トークンが返した値を参照するように、サーチ文字列を変更します。ダッシュボード内の見出しや動的ドリルダウン内でトークンを参照することもできます。
   「サーチ内でのトークンの参照」および「フォームへの時間入力の追加」を参照してください。
4. フォームを表示するサーチの実行時期などの、デフォルトのフォーム動作を指定します。
   「フォームの動作の指定」を参照してください。
5. 複数のオプションがある入力の場合、ユーザーオプションを指定します。
   これは、ドロップダウン、チェックボックス、複数選択、およびラジオ入力に適用されます。
   「複数のオプションを受け付ける入力の選択肢の指定」を参照してください。
6. 入力のその他の属性を指定します。
   各入力タイプの例については、「フォーム入力の例」を参照してください。
7. (オプション) 入力とパネルを再配置します。
   パネルとフォーム入力の配置をドラッグアンドドロップすることができます。また、入力を特定のパネルにドラッグすることもできます。

フォームを作成するための入力の追加

既存のダッシュボードに入力を追加すると、ダッシュボードがフォームに変換されます。入力を追加すると、そのベースとなるシンプル XML のトップレベルのエレメントが、<dashboard> から <form> に変化します。

1. ダッシュボードから、**[編集]** > **[パネルの編集]** をクリックします。
2. **[入力の追加]** をクリックします。
3. 入力のリストから、入力を選択します。
   たとえば、**[テキスト]** を選択して、ユーザー入力にテキストフィールドを追加します。
4. (オプション) 複数の入力がある場合、入力をドラッグしてフォーム上の配置を変更することができます。
5. (オプション) パネルに入力をドラッグします。
   トークンを使って、入力をそのパネル内の視覚エフェクトにのみ適用するように設定することができます。
   「フォーム入力のトークン」を参照してください。
6. 入力を編集するには、入力の **[編集]** アイコンを選択して、属性を編集します。
   詳細は、以下のセクションを参照してください。
7. 入力の編集が完了したら、**[完了]** をクリックします。

フォーム入力のトークン

フォームに入力を追加すると、ダッシュボードエディタはその入力で使用する一意のトークンを生成します。このトークンは、動的な値のプレースホルダです。一般的にはサーチ文字列内でトークンを参照して、フォームに表示する返された値を指定します。また、パネルのヘッダーおよびグラフのドリルダウン内のトークンを参照することもできます。

時間入力とトークンを使用することで、フォーム内の時間入力を識別します。その時間入力で表示する各パネルに対して、パネルのサーチがそのトークンによる時間入力を参照するように変更します。

以下の例は、フォーム内でトークンを参照する代表的な 2 種類の例を表しています。

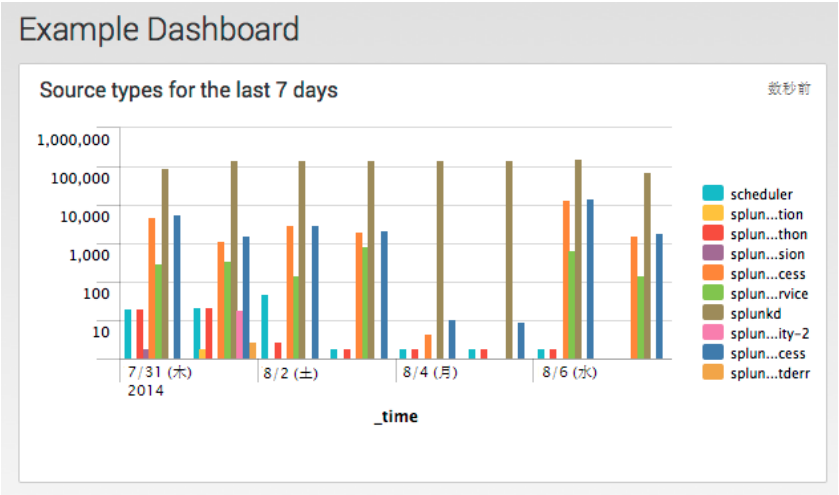
サーチ内のトークンの参照
フォームへの時間入力の追加

また、グラフのドリルダウン機能を使用する場合に、トークンを参照することもできます。「動的ドリルダウン」を参照してください。

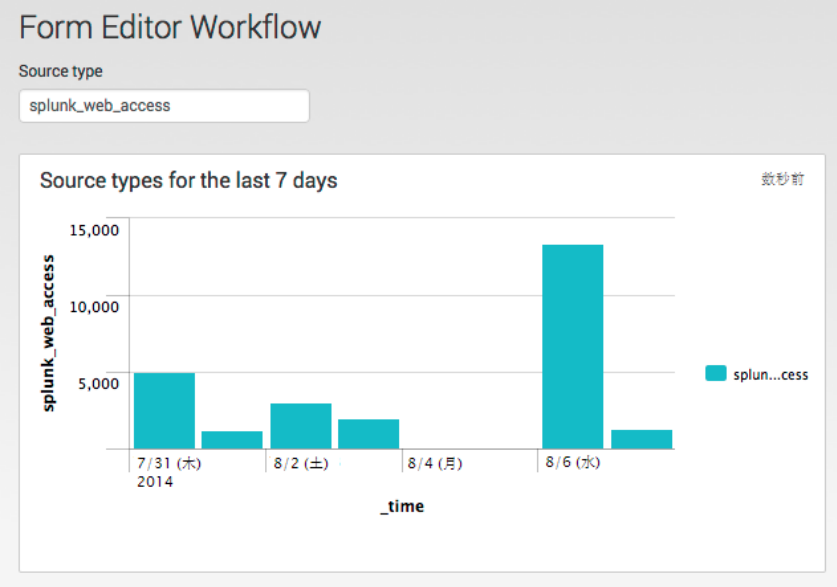
サーチ内のトークンの参照

以下のダッシュボード内のパネルには、ソースタイプの時間グラフを表示する、次のサーチが含まれています。入力を追加してダッシュボードをフォームに変換する場合、サーチ内のトークンを参照してフォームに返す値を指定します。

```
index=_internal | timechart count by sourcetype
```



1. ダッシュボードから、**[編集] > [パネルの編集]** をクリックします。
2. ダッシュボードをフォームに変換するための、テキスト入力を追加します。
   ダッシュボードエディタは、トークン属性のために以下の一意の値を生成します：`field[n]`。
3. トークン用に生成された値を、より分かりやすい `src_type_tok` に変更します。
   ダッシュボード内で、トークンの値は一意でなければなりません。
4. パネル内のサーチ文字列を編集して、トークンを参照します。
   ```
   index=_internal sourcetype=$src_type_tok$ | timechart count by sourcetype
   ```
5. **[保存]** をクリックします。**[完了]** をクリックします。
   これでフォームには、サーチが返す結果をフィルタリングするテキスト入力が追加されました。

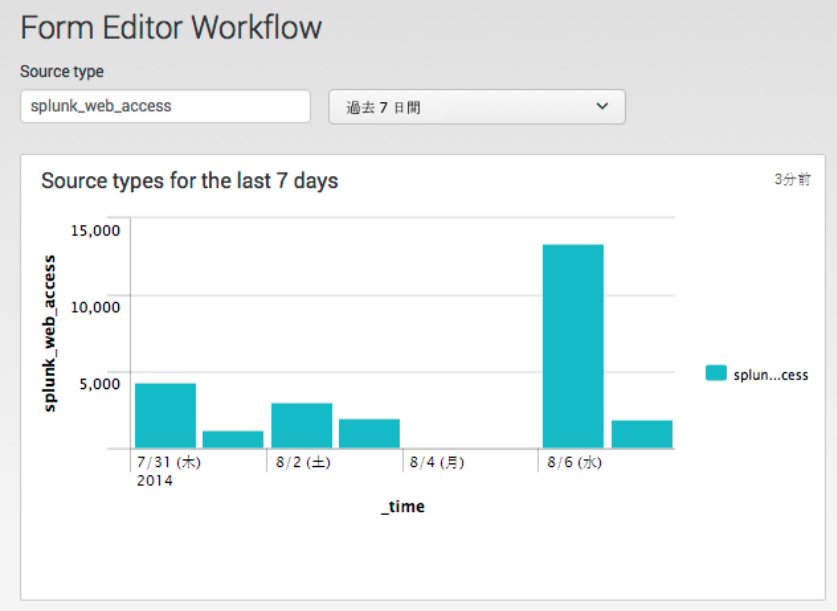


フォームへの時間入力の追加

以下の例は、フォームへの時間入力の追加方法を表しています。最初にフォームにパネルを追加する際に、パネルの時間範囲を指定します。新たに追加した時間入力からの値を適用するには、パネルのサーチを編集して、パネルの時間範囲に優先する設定を行います。

1. ダッシュボードから、**[編集] > [パネルの編集]** をクリックします。
2. **[入力の追加] > [時間]** を選択します。
3. 入力の**編集アイコン**を選択し、タイムレンジピッカーの属性を指定します。
   (オプション) トークンに対して、分かりやすい名前を指定します。たとえば、「time_range_7days_tok」のように指定します。
4. **[適用]** をクリックします。
5. 時間入力を適用する各パネルに対して、以下の作業を行います。

   a. **サーチアイコン**をクリックします。
   b. [サーチ文字列の編集] をクリックします。
   c. **[時間範囲]** で、**[共有タイムピッカー]** を選択します。
      時間範囲の選択の詳細は、「フォーム内の複数の時間入力」を参照してください。
   d. **[保存]** をクリックします。
6. **[完了]** をクリックします。



フォーム内の複数の時間入力

フォームに 1 つまたは複数の時間入力を配置して、視覚エフェクトの時間範囲を指定することができます。また、タイムピッカーを参照せずに、シンプル XML コードに直接視覚エフェクトの時間範囲を指定することもできます。

フォームにタイムピッカーを追加すると、ダッシュボードエディタはそのタイムピッカーで使用する一意のトークンを生成します。生成されたトークン名を、より分かりやすい名前に変更することもできます。視覚エフェクトはトークンを参照して、サーチの時間範囲を適用します。

タイムピッカー用のトークンを指定しない場合、タイムピッカーはグローバルになります。タイムピッカートークンの参照またはコード内への直接の指定による時間範囲を指定しない視覚エフェクトは、グローバルタイムピッカーから時間範囲を適用します。

時間入力を指定する方法を以下に示します。

- グローバルタイムピッカー
  時間入力ではトークンを指定しません。
- タイムピッカーのトークン参照
  視覚エフェクトはタイムピッカーのトークンを参照します。
- サーチに時間を直接指定
  シンプル XML コードの <earliest> および <latest> エレメントで時間範囲を指定します。

視覚エフェクトの時間範囲の選択

パネルエディタで時間範囲を選択します。

1. ダッシュボードから、**[編集]** > **[パネルの編集]** を選択します。
2. パネルの**サーチアイコン**をクリックします。
3. **[サーチ文字列の編集]** を選択します。
4. **[時間範囲]** ドロップダウンメニューから、タイムピッカーを選択します。
   フォーム内のタイムレンジピッカーに応じて、以下のいずれかを選択します。
   - **共有タイムピッカー (グローバル)**
     トークン参照なしのタイムピッカーです。
   - **共有タイムピッカー (トークン名)**
     トークンで参照されるタイムピッカー。
   - **明示的選択**
     シンプル XML コード内に直接時間範囲を指定します。
   - **トークン**
     時間範囲の**もっとも早い**および**もっとも遅い**値を表すトークンを指定します。
5. [**保存**] をクリックします。**[完了]** をクリックします。

フォームの動作の指定

フォームにサーチからの結果を設定するためのトークンの送信にはさまざまな方法があります。トークン値の送信方法により、フォームのサーチからの結果の表示時期が決まります。

ページ読み込み時のトークン値送信

ページ読み込み時にトークン値を送信するには、自動実行動作を有効にします。

1. ダッシュボードから、**[編集]** > **[パネルの編集]** をクリックします。
2. **[ダッシュボードの自動実行]** チェックボックスを有効にします。**[完了]** をクリックします。

入力変更時のトークン値送信

各入力に対して、入力が変更された場合にトークン値を送信することができます。デフォルトで、この動作は有効になっています。**[変更時にサーチ]** を有効にした場合、フォームに [送信] ボタンは必要ありません。入力変更時のサーチを有効または無効にする手順を以下に示します。

1. ダッシュボードから、**[編集]** > **[パネルの編集]** をクリックします。
2. 編集する入力を選択します。
3. **[変更時にサーチ]** チェックボックスを選択または選択解除します。**[完了]** をクリックします。

入力値送信のための [送信] ボタンの追加

ユーザーがボタンをクリックした時に入力値を送信する、[送信] ボタンをフォームに追加できます。一般的に、[送信] ボタンを使用する場合、入力の **[変更時にサーチ]** プロパティは指定しません。

ダッシュボードエディタは、[送信] ボタンをすべてのフォーム入力の右側に配置します。[送信] ボタンの位置を変更することはできません。

1. ダッシュボードから、**[編集]** > **[パネルの編集]** をクリックします。
2. **[入力の追加]** > **[送信]** を選択します。
3. フォームの編集を継続します。編集が完了したら、**[完了]** をクリックします。

シードとデフォルト値について

フォーム入力のシードとデフォルト値を指定することができます。

- **デフォルト**
  ユーザーが入力値を指定しない場合の入力値です。

- **シード**
  (テキスト入力のみ) ページ読み込み時の入力の初期値として、指定された値を使用します。シード値は、初期のページ読み込み時にのみ適用されます。テキスト入力フィールドが空の場合、シード値は適用されません。

  注意：シードとデフォルトの主な違いは、シード値の場合ユーザーが空文字列を送信できることにあります。シード値とデフォルト値の両方を指定した場合は、デフォルト値のみが適用されます。

複数のオプションを受け付ける入力の選択肢の指定

さまざまな葉フォーム入力が、複数の選択項目を提供しています。選択項目の指定手順は、これらの各フォーム入力で同じです。そのため、ある入力タイプから別の入力タイプに手軽に変更することができます。

以下の入力で、複数の選択オプションを提供できます。

チェックボックス
ドロップダウン
複数選択
ラジオボタン

注意：複数選択およびチェックボックス入力オプションには、複数値を指定するための追加手順が必要です。「チェック ボックスおよび複数選択の複数値の指定」を参照してください。

オプションに静的な (ハードコード化された) ラベルと値を指定、または動的にラベルと値を生成するサーチを指定することができます。

静的オプションによる選択項目の指定

複数のオプションを含む入力に対する、静的オプションの指定手順を以下に示します。以下の手順はドロップダウン入力を想定していますが、以下の入力タイプにも適用されます。

- ドロップダウン
- チェックボックス
- 複数選択
- ラジオボタン

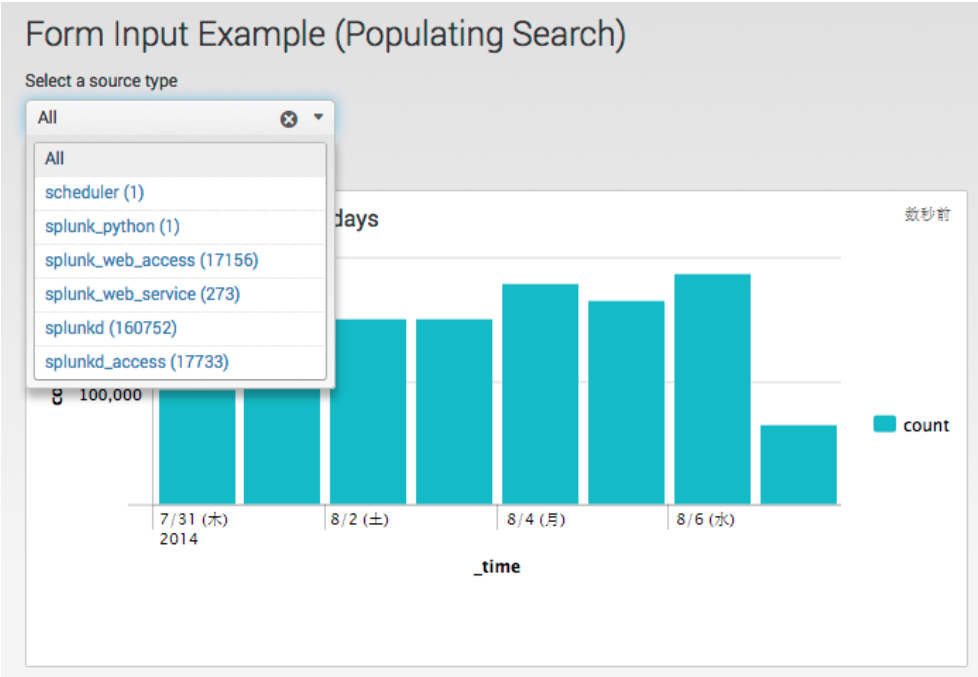
1. ダッシュボードから、**[入力の追加]** > **[ドロップダウン]** を選択します。
2. 入力の編集アイコンを選択します。**[静的オプション]** を選択します。
3. 最初のオプションの名前と値を指定します。
4. 他の各オプションに対して、**[オプションの追加]** をクリックして、そのオプションの名前と値を指定します。
5. (オプション) **[デフォルト]** フィールドに移動して、デフォルト値を指定します。
6. (オプション) オプションをドラッグ アンド ドロップして再配置します。
7. **[適用]** をクリックします。

動的オプションによる選択項目の指定

複数のオプションを持つ入力に対して、サーチを使ってラベルと値を指定することができます。以下の手順はドロップダウン入力を想定していますが、以下の入力タイプにも適用されます。

- ドロップダウン
- チェックボックス
- 複数選択
- ラジオボタン

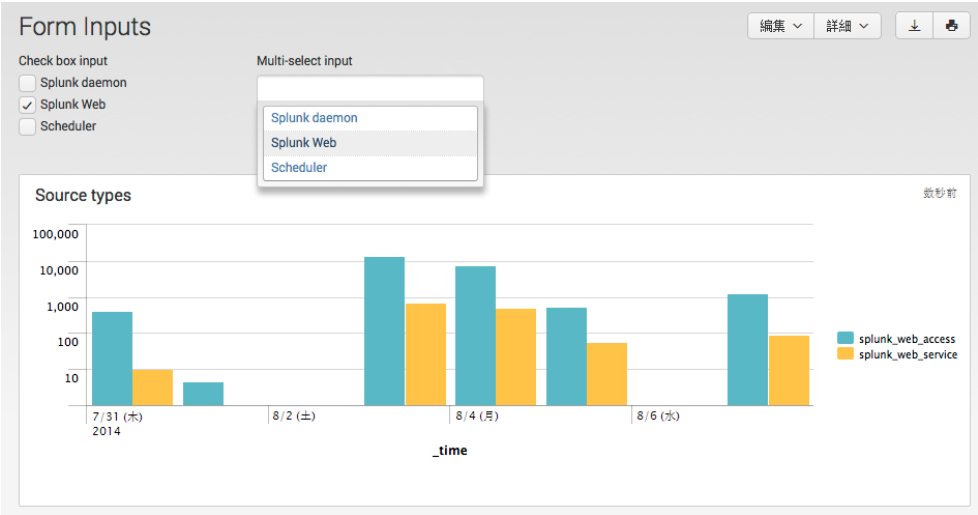
ドロップダウンメニューのオプションを設定するサーチの作成手順を以下に示します。これはオプション用に生成されたラベルをカスタマイズするための、1 つの方法を表しています。



1. 以下のサーチを使用するパネルを持つダッシュボードを作成します。
   ```
   index=_internal $source_tok$ | timechart count
   ```
2. ダッシュボードから、**[入力の追加]** > **[ドロップダウン]** を選択します。
3. 入力の**編集アイコン**を選択します。入力エディタで以下のフィールドを編集します。
   - **ラベル**：Select a source type
   - **変更時にサーチ**：有効
   - **トークン**： source_tok
   - **トークンプリフィックス**： sourcetype="
   - **トークンサフィックス**："
   - (静的オプション) **名前**：All; **値**：*
4. **[動的オプション]** をクリックして、設定サーチを指定します。
5. メニューオプションを設定する、以下のインラインサーチを指定します。
   ```
   index=_internal | stats count by sourcetype | eval label=sourcetype." (".count.")"
   ```
   このサーチは eval 関数を使って、ラベルフィールドを定義しています。
6. (オプション) **[サーチの実行]** を選択して、サーチ結果をプレビューします。
7. 設定用のサーチから返されたフィールドを指定します。
   ラベルのフィールド：label
   値のフィールド：sourcetype
8. **[適用]** をクリックします。

チェック ボックスおよび複数選択の複数値の指定

複数選択およびチェックボックスフォーム入力は、ユーザーが複数のオプションから選択できるという点で他の入力と異なります。パネル内でモデル化するソースタイプを選択するための、チェックボックス入力と複数選択入力を以下の図に示します。



上記のフォームでソースタイプを指定するには、返す値を示すサーチ文字列を作成します。この例では、次のサーチ文字列により、ソースタイプの複数値選択が可能になります。

```
(sourcetype="splunkd" OR sourcetype="splunk_web_access" OR sourcetype="splunkd_access")
```

パネルが使用するサーチは、チェック ボックスおよび複数選択のトークン値に、他のフォーム入力とは異なる方法でアクセスします。トークンに送信された修飾子を使用します。

```
index=_internal $src_type_tok$ | timechart count by sourcetype
```

入力エディタには、チェックボックスまたは複数選択用の複数の選択項目値を指定するための編集フィールドが用意されています。以下のテーブルは、これらのフィールドと以下のサーチ文字列を作成するサンプル値を記載しています。

```
(sourcetype="<b>selected value</b>" OR sourcetype="<b>selected value</b>" OR ... )
```

エディタのフィールド	説明	値の例
トークンプリフィックス	入力エレメントの値の先頭に付ける文字列。 複数の選択項目の場合、一般的にこれは値を選択する文字列を囲む左括弧になります。	(
トークンサフィックス	入力エレメントの値の最後に追加する文字列。 複数の選択項目の場合、一般的にこれは値を選択する文字列を囲む右括弧になります。	)
トークン値プリフィックス	入力エレメントの値の先頭に付ける文字列。正規表現を使用できます。デフォルト値は左二重引用符になります。 一般的にこれは、複数値を選択するサブ文字列の開始部になります。	sourcetype="
トークン値サフィックス	入力エレメントの値の最後に追加する文字列。正規表現を使用できます。デフォルト値は右二重引用符になります。 一般的にこれは、複数値を選択するサブ文字列の終了部になります。	"
区切り文字	選択した各値の間に配置する文字列です。一般的には「 OR 」や「 AND 」 を大文字で指定します。引用符は指定しないでください。ただし、文字列の前後にはスペースを指定する必要があります。 デフォルト値：" " デフォルト値に引用符は含まれていません。引用符は、デフォルト値がスペースであることを示すために、便宜的に用いられています。	OR

チェック ボックスまたは複数選択入力で、複数の選択項目を有効にする方法を以下の手順に示します。

1. ダッシュボードから、**[編集] > [パネルの編集]** をクリックします。
2. **[入力の追加]** を選択します。**[チェックボックス]** または **[複数選択]** を選択します。
3. **[ラベル]**、**[変更時にサーチ]**、および **[トークン]** を指定します。
4. 「静的オプションによる選択項目の指定」および「動的オプションによる選択項目の指定」の説明に従って、選択項目を指定します。
5. 上記のテーブルの編集フィールドを使って、複数選択サーチ文字列を作成します。
   (推奨) プレビュー機能を使って、複数選択サーチ文字列を確認します。
6. (オプション) デフォルト値を指定します。
7. **[適用]** をクリックします。**[完了]** をクリックします。

フォーム入力の例

このセクションは、各フォーム入力の例、および例を実装するための主要フィールドの一覧を記載しています。

チェックボックス

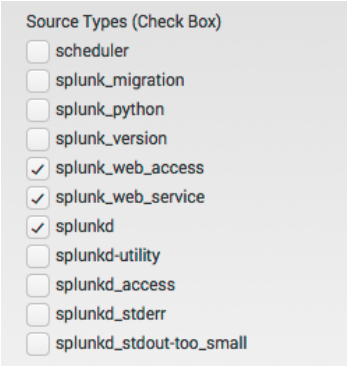
この例は、チェックボックス入力を使って時間グラフに表示するソースタイプを指定します。設定用のサーチは、選択する利用可能なオプションを示しています。デフォルトでは、3 つのソースタイプが選択されています。

```
splunk_web_access
splunk_web_service
splunkd
```

この例では、**[変更時にサーチ]** を有効にします。選択が行われた場合に、フォームが読み込まれます。

パネルのデフォルトの縦棒グラフには、以下のベースサーチの結果が表示されます。視覚エフェクトは、**[トークン]** に指定された値を使って、入力値を参照します。この例では、トークン名は `src_type_tok` になります。

```
index=_internal $src_type_tok$ | timechart count by sourcetype
```



Src types checkbox panel.png

全般設定

入力の **[ラベル]** と **[変更時にサーチ]** 動作を指定します。この例では、[変更時にサーチ] を有効にします。

トークンオプション

チェックボックス入力が返す値を指定するには、**[トークンオプション]** を使用します。

[トークン] フィールドの場合、値を返すトークンの名前を指定します。視覚エフェクトのベースサーチは、このトークンを参照します。この例では、`src_type_tok` を指定します。

返された値のサーチを作成するには、次のフィールドを使用します。入力エディタの **[プレビュー]** フィールドは、これらのフィールドの編集に伴って更新されます。

- **トークンプリフィックス**
- **トークンサフィックス**
- **トークン値プリフィックス**
- **トークン値サフィックス**
- **区切り文字**

以下の表に記載されている値の例は、以下のサーチ文字列を構築します。

(sourcetype="splunkd" OR sourcetype="splunk_web_access" OR ...)

動的にチェックボックスを作成したら、**[デフォルト]** フィールドから、デフォルトで有効になっているチェックボックスを選択します。

静的オプション

入力のチェックボックスの **[名前]** と **[値]** を明示的に定義するには、静的オプションを使用します。

この例では、静的オプションを空のままにしています。設定用サーチを使って、入力のチェックボックスを定義します。

動的オプション

入力のチェックボックスを定義するために、レポートを参照するか、または設定用インラインサーチを定義します。

この例では、以下のインラインサーチを使用します。

```
| metadata type=sourcetypes index=_internal
```

この例は、全時間に対してサーチを実行します。

チェックボックスの名前と値のペアを指定するために、フィールド名を使用します。この例では、**[ラベルのフィールド]** および **[値のフィールド]** の両方に対して、`sourcetype` フィールドを指定します。

チェックボックス入力の値の例

以下の表に、チェックボックス入力の値の例を示します。

エディタのフィールド	値の例
全般	
ラベル	ソースタイプ (チェックボックス)
変更時にサーチ	*有効*
トークンオプション	
トークン	src_type_tok
デフォルト	splunk_web_access splunk_web_service splunkd
トークンプリフィックス*	(
トークンサフィックス*	)
トークン値プリフィックス*	sourcetype="
トークン値サフィックス*	"
区切り文字*	OR
動的オプション	
コンテンツタイプ	インラインサーチ
サーチ文字列	`| metadata type=sourcetypes index=_internal`
時間範囲	全時間
ラベルのフィールド	sourcetype
値のフィールド	sourcetype
* これらのフィールドは、動的にチェックボックスを作成するサーチ文字列を構築します。区切り文字フィールドでは、文字列の左右にスペースを忘れずに指定してください。	

ドロップダウン入力

この例はドロップダウン入力を使って、時間グラフとして表示するソースタイプを指定します。結果を横棒グラフとして表示するパネルには、以下のベースサーチを使用しています。

```
index=_internal sourcetype=$src_type_tok$ | timechart count by sourcetype
```

トークン `$src_type_tok$` は、ドロップダウンが指定する値を参照しています。

Src types dropdown panel.png

この例では、静的オプションを使って、ドロップダウンの選択項目を定義しています。

この例は、**[トークンプリフィックス]** に `splunk` を指定しています。選択した各値が、値の前にトークンプリフィックスとして付けられます。

ドロップダウンには、デフォルト値があります。

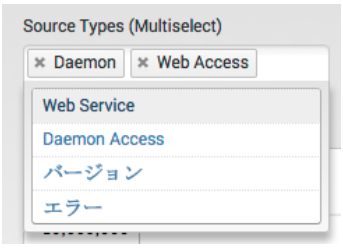
この例は、[送信] ボタンを使ってサーチを実行します。選択内容を変更しても、[送信] ボタンをクリックするまでは適用されません。

エディタのフィールド	値の例
全般	
ラベル	ソースタイプ (ドロップダウン)
変更時にサーチ	*未指定*
トークンオプション	
トークン	src_type_tok
デフォルト	Daemon
トークンプリフィックス	Splunk
静的オプション	
名前：値	Daemon : d
名前：値	Web Service ：_web_service
名前：値	Web Access :_web_access
名前：値	Daemon Access : d_access

複数選択

この例は、複数選択入力を使って時間グラフに表示するソースタイプを指定します。パネルのデフォルトの縦棒グラフには、以下のベースサーチの結果が表示されます。

```
index=_internal $src_type_tok$ | timechart count by sourcetype
```



Src types multiselect panel.png

この例では、静的オプションを使って、ドロップダウンの選択項目を定義しています。

デフォルトでは、2 つのソースタイプが選択されています。

Daemon
Web Access

この例では、**[変更時にサーチ]** を有効にします。選択が行われた場合に、フォームが読み込まれます。

複数選択入力の場合、以下のサーチ文字列を作成して、複数の選択値を定義します。

(sourcetype="splunkd" OR sourcetype="splunk_web_access" OR ...)

トークン `$src_type_tok$` は、パネルの内容を表示するサーチ内の、このサーチ文字列を参照します。サーチ文字列を作成するフィールドを以下の表に示します。

エディタのフィールド	値の例
全般	
ラベル	ソースタイプ (複数選択)
変更時にサーチ	*有効*
トークンオプション	
トークン	src_type_tok
デフォルト	Daemon WEb Access
トークンプリフィックス*	(
トークンサフィックス*	)
トークン値プリフィックス*	sourcetype="
トークン値サフィックス*	"
区切り文字*	または
静的オプション	
名前：値	Daemon : splunkd
名前：値	Web Service : splunk_web_service
名前：値	Web Access : splunk_web_access
名前：値	Daemon Access : splunkd_access
名前：値	Daemon Access : splunkd_access
名前：値	Version : splunk_version
名前：値	Error : splunkd_stderr
* これらのフィールドは、トークン値を提供するサーチ文字列を構築します。区切り文字フィールドでは、文字列の左右にスペースを忘れずに指定してください。	

ラジオボタン入力

この例はラジオボタン入力を使って、時間グラフとして表示するソースタイプを指定します。結果を面グラフとして表示するパネルには、以下のベースサーチを使用しています。

```
index=_internal sourcetype=$src_type_tok$ | timechart count by sourcetype
```

Src types radio panel.png

トークン `$src_type_tok$` は、ドロップダウンが指定する値を参照しています。

この例では、静的オプションを使って、ドロップダウンの選択項目を定義しています。

ラジオボタン入力には、デフォルト値があります。

この例では、**[変更時にサーチ]** を有効にします。選択が行われた場合に、フォームが読み込まれます。

エディタのフィールド	値の例
全般	
ラベル	ソースタイプ (ラジオ)
変更時にサーチ	*有効*
トークンオプション	
トークン	src_type_tok
デフォルト	Web Service
静的オプション	
名前：値	Daemon : splunkd
名前：値	Web Service : splunk_web_service
名前：値	Web Access : splunk_web_access
名前：値	Daemon Access : splunkd_access

テキスト入力

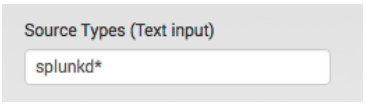
この例はテキスト入力を使って、時間グラフとして表示するソースタイプを指定します。結果を円グラフとして表示するパネルには、以下のベースサーチを使用しています。

```
index=_internal sourcetype=$src_type_tok$ | timechart count by sourcetype
```

トークン `$src_type_tok$` は、テキスト入力に指定されている値を参照しています。

この例は、デフォルト値を指定せずに、シード値 `splunkd*` を指定します。初期読み込み時にシード値が適用されます。フォームは、新しい値を指定した時に再読み込みされます。

デフォルト値がないため、テキスト入力が空の場合結果は返されません。



Src types text panel.png

エディタのフィールド	値の例
全般	
ラベル	ソースタイプ (テキスト入力)
変更時にサーチ	*有効*
トークンオプション	
トークン	src_type_tok
デフォルト	*未指定*
シード	splunkd*

時間入力

この例は、フォーム内のパネルの時間範囲を指定するための、時間入力の使用方法を表しています。フォームには、時間グラフとして表示するソースタイプを示す、ラジオボタン入力が含まれています。結果を縦棒グラフとして表示するパネルには、以下のベースサーチを使用しています。

```
index=_internal sourcetype=$src_type_tok$ | timechart count by sourcetype
```

Src types time panel.png

この例は、パネル内の時間入力を参照するために `time_input_tok` を指定しています。

パネルエディタで、**[サーチ文字列の編集]** を選択します。**[時間範囲]** ドロップダウンから、**[共有タイムピッカー (time_input_tok)]** を選択します。

時間入力のデフォルト値は、**[過去 7 日間]** です。

この例では、時間入力の **[変更時にサーチ]** を有効にします。フォームは、新しい時間範囲が選択されると読み込まれます。

エディタのフィールド	値の例
全般	
ラベル	時間入力
変更時にサーチ	*有効*
トークンオプション	
トークン	time_input_tok
デフォルト	過去 7 日間

視覚エフェクトの編集

ビジュアルエディタには、視覚エフェクトを作成、変更するための、さまざまな機能が用意されています。データを表示するために選択する視覚エフェクトは、サーチが返す結果とそれをどのように表現したいかによって異なります。

ここでは、ダッシュボードエディタおよび [サーチ] ページからのの視覚エフェクトの編集方法について説明していきます。またピボットエディタを使って視覚エフェクトを編集することもできます。ピボットエディタは、ピボット内の定義を視覚エフェクトのプロパティに対応させるために、ビジュアルエディタよりも多くの視覚エフェクト定義オプションを提供しています。詳細は、「ピボットエディタによるピボットグラフと視覚エフェクトの設計」を参照してください。

ビジュアルエディタについて

ビジュアルエディタは、Splunk Web の以下の場所から利用することができます。

- ダッシュボード編集時のパネルエディタ。
- 変換サーチ実行後のサーチ ([視覚エフェクト] タブで)。

ビジュアルエディタは単一のエディタではなく、視覚エフェクトを選択、変更するための、一連の編集ダイアログから成り立っています。利用できる編集機能は、パネルエディタからの視覚エフェクトを編集するのか、またはサーチからの視覚エフェクトを編集するのかによって異なります。

編集ダイアログの内容も、サーチ結果やそれらの結果に対して選択された視覚エフェクトによって異なります。

ビジュアルエディタには、以下のダイアログが含まれています。

- **視覚エフェクトピッカー**
  ドロップダウンリストから視覚エフェクトを選択します。ソースサーチに基づいて適切な視覚エフェクトをいくつか選択することをお勧めします。利用できる視覚エフェクトの一部は、サーチ結果によっては適切ではないこともあります。

  視覚エフェクトピッカーが示すアイコンが、現在選択されている視覚エフェクトを表しています。

- **視覚エフェクトの書式設定**
  選択した視覚エフェクトのオプションを指定することができます。利用できるオプションは、視覚エフェクトによって異なります。

- **サーチエディタ**
  パネルエディタからのみ利用できます。パネルの結果を生成するベースサーチまたはレポートを変更することができます。

ビジュアルエディタの起動

ビジュアルエディタの起動方法は、ダッシュボードを編集するのか、またはサーチの結果を表示するのかによって異なります。

ダッシュボードエディタ

ダッシュボードエディタでは、書き込み**権限**を持つダッシュボード内のパネルのみを編集することができます。ダッシュボードに対して読み取り専用アクセス権しかない場合、ダッシュボード内のパネルの外観を変更することはできません。デフォルトでは、自分がダッシュボードエディタを使って作成した任意のダッシュボードに対して、書き込み権限を保有しています。ただし、管理権限を持つユーザーは、アクセス権を変更することができます。

ダッシュボード内のパネルに対してビジュアルエディタを使用するには：

1. ダッシュボードで、**[編集] > [パネルの編集]** を選択して、ダッシュボードの編集を有効にします。
2. 編集するパネルに対して、適切なアイコンを選択します。

   - **[パネルのプロパティ] アイコン**：パネルのベースサーチを編集します。
   - **[視覚エフェクト] アイコン**：視覚エフェクトを変更します。
   - **[視覚エフェクトの書式設定] アイコン**：選択した視覚エフェクトを設定します。

[サーチ] ページ

[サーチ] ページで、サーチ結果を取得した後にビジュアルエディタを開きます。「変換コマンドについて」で説明しているように、サーチは変換サーチでなければなりません。

サーチを実行したら、視覚エフェクトを編集するために適切なアイコンを選択します。

- **[視覚エフェクト] アイコン**：視覚エフェクトを変更します。
- **[視覚エフェクトの書式設定] アイコン**：選択した視覚エフェクトを設定します。

視覚エフェクトの編集

各視覚エフェクトには、変更可能な一連の設定プロパティが含まれています。多くのグラフが同じプロパティを共有していますが、一部のプロパティは特定の種類のグラフにのみ適用されます。ここでは、編集に利用できるグラフのプロパティを簡単に説明していきます。

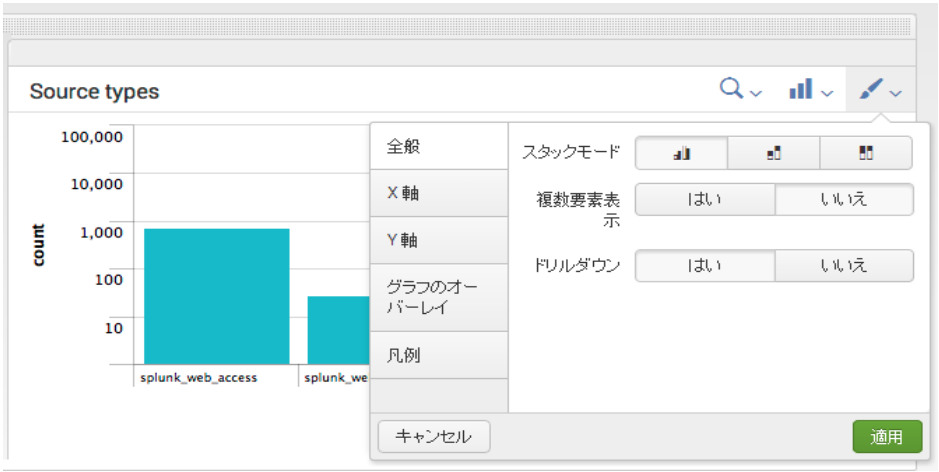
視覚エフェクトのプロパティの詳細は、以下の項目を参照してください。

- 視覚化リファレンス
- グラフ設定リファレンス

全般

全般プロパティは、視覚エフェクトによって異なります。

面、横棒、縦棒、折れ線グラフ



- **スタックモード** (縦棒、横棒、および面)
  グラフ内のデータの表現方法。オプションを以下に示します。

  スタックなし
  スタック
  100% スタック100%

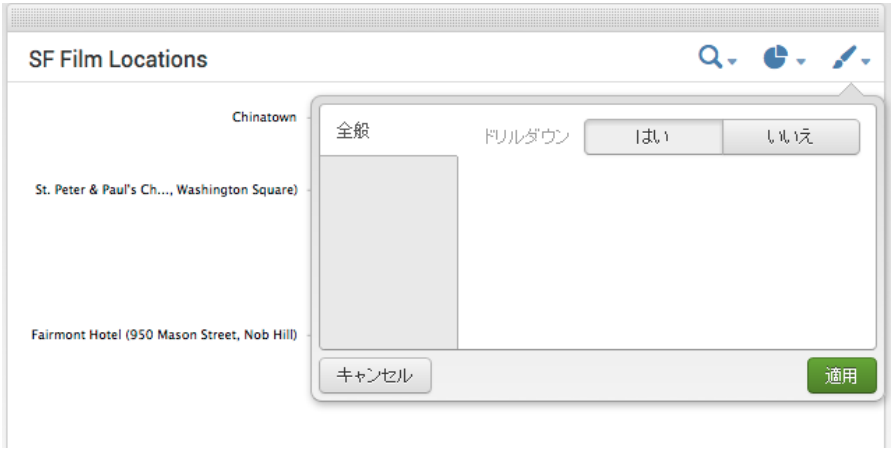
- **ドリルダウン**
  視覚エフェクトの**ドリルダウン**機能を有効にします。テーブルの場合、行またはセルに対するドリルダウンを有効にできます。詳細は、「基本的なテーブル/グラフのドリルダウンアクションの概要」を参照してください。

- **複数シリーズモード**
  複数シリーズモードを有効または無効にします。

- **Null 値** (面、折れ線)
  存在していない Y 軸値の表現方法を指定します。オプションを以下に示します。

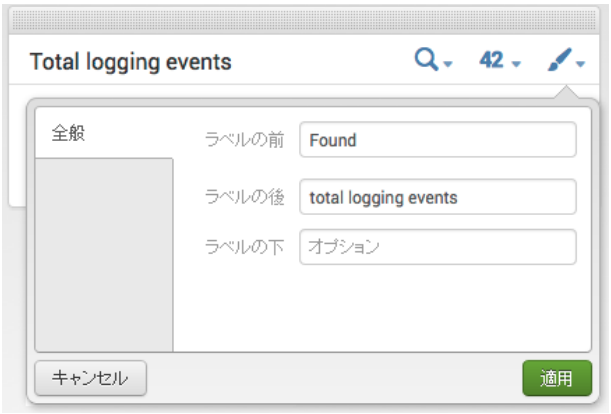
  ギャップ：値が 0 未満のデータポイントには接続しない、または次のデータポイントにのみ接続します。
  0：欠損値のデータポイントを 0 に接続します。
  接続：次のデータポイントに接続します。

円グラフ、散布図



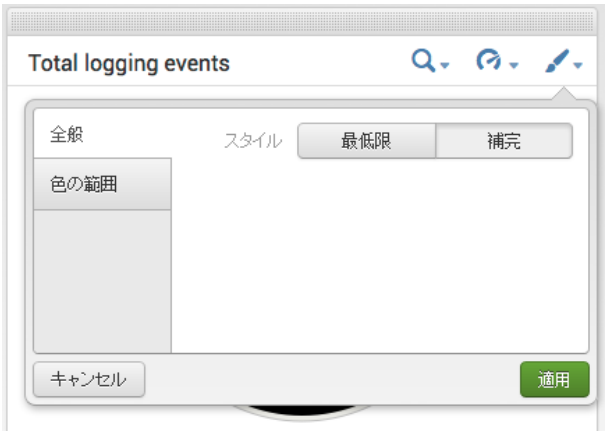
- **ドリルダウン**
  視覚エフェクトの**ドリルダウン**機能を有効にします。詳細は、「ドリルダウン動作」を参照してください。

単一値



- **ラベルの前**：値の前に表示するテキスト。
- **ラベルの後**：値の後に表示するテキスト。
- **ラベルの下**：値の下に表示するテキスト。

フィラー、マーカー、および放射状ゲージ



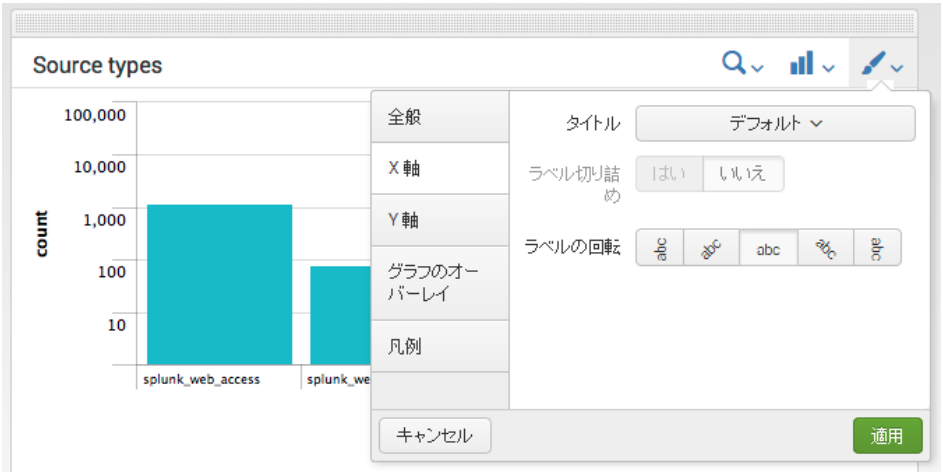
- **スタイル**：

  最低限：ゲージの基本版です。
  補完：クロム、シェード、その他の機能などの、現実世界のゲージをモデルにした、グラフィック的にスタイル化されたゲージ。

X 軸

グラフの X 軸のプロパティを指定します。

面、横棒、縦棒、折れ線グラフ、および散布図



- **タイトル**：X 軸のタイトルを指定します。以下のいずれかを選択します。

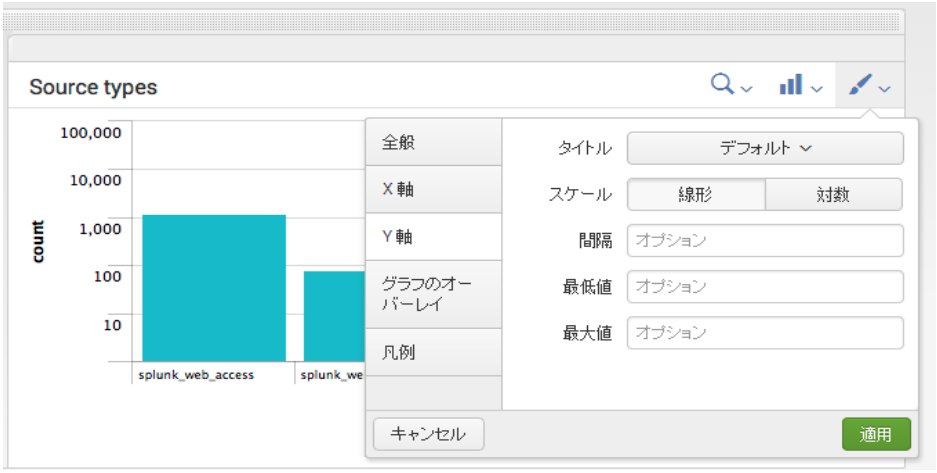
  デフォルト：パネルのサーチから得られたタイトル。
  カスタム：テキストボックスにカスタムタイトルを入力します。
  なし：タイトルを表示しません。

- **ラベルの回転**：列の下へのラベルの表示方法を設定します。

Y 軸

グラフの Y 軸のプロパティを指定します。

面、横棒、縦棒、折れ線グラフ、および散布図



- **タイトル**：Y 軸のタイトルを指定します。以下のいずれかを選択します。

  デフォルト：パネルのサーチから得られたタイトル。
  カスタム：テキストボックスにカスタムタイトルを入力します。
  なし：タイトルを表示しません。

- **スケール**：**[線形]** または **[対数]** を選択します。[対数] は、ピーク値が非常に大きな場合に、それを縮小表示するために役立つ対数スケールを使用します。
- **間隔**：軸目盛間の単位を入力します。
- **最低値**：表示する最低値。最低値未満の値はグラフには表示されません。
- **最大値**：表示する最大値。最大値を超える値はグラフには表示されません。

グラフのオーバーレイ

グラフのオーバーレイのプロパティを指定します。詳細は、「グラフのオーバーレイ」を参照してください。

面、縦棒、折れ線グラフ

- **オーバーレイ**：オーバーレイとして表示するフィールドを選択します。
- **軸として表示**：2 番目の Y 軸にオーバーレイする場合は、**[オン]** を選択します。
- **タイトル**：オーバーレイのタイトルを指定します。
- **スケール**：**[継承]**、**[線形]**、または **[対数]** を選択します。[継承] はベースとなるグラフのスケールを使用します。[対数] は、とても大きなピーク値を縮小表示する場合に役立つ対数スケールを使用します。
- **間隔**：軸目盛間の単位を入力します。
- **最低値**：表示する最低値。最低値未満の値はグラフには表示されません。
- **最大値**：表示する最大値。最大値を超える値はグラフには表示されません。

凡例

グラフの凡例のプロパティを指定します。

面、横棒、縦棒、折れ線グラフ、散布図

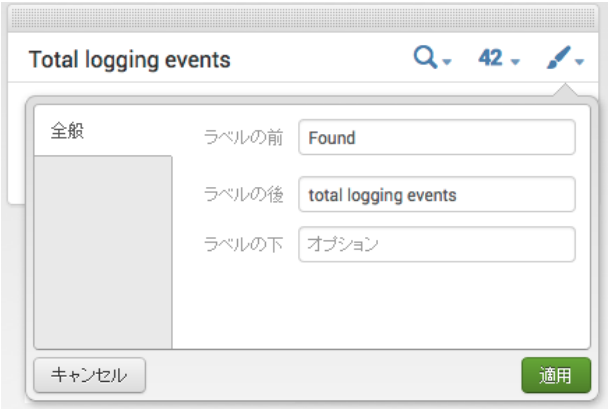
- **位置**：視覚エフェクト内の凡例の配置場所 (または凡例を表示しない)。
- **切り詰め**：表示するには長すぎる名前の表現方法。

  最後を切り捨て：後方の文字列を省略します。
  中央を切り捨て：中央の文字列を省略します。
  最初を切り捨て：先頭の文字列を省略します。

単一値とゲージ

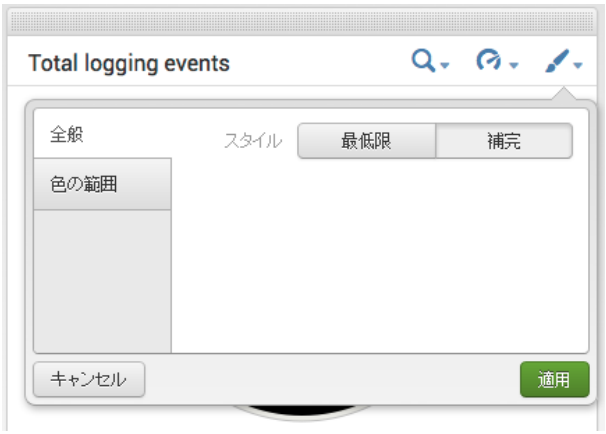
単一値として返される値向けに、Splunk には単一値視覚エフェクトおよび各種ゲージ (放射状、フィラー、マーカー) が用意されています。これらの視覚エフェクトに対しては、以下の機能を指定することができます。

単一値 (一般)



- **ラベルの前**：値の前に表示するテキスト。
- **ラベルの後**：値の後に表示するテキスト。
- **ラベルの下**：値の下に表示するテキスト。

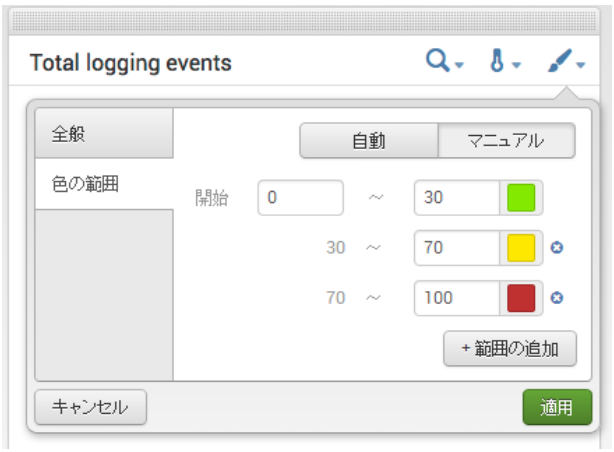
フィラー、マーカー、および放射状ゲージ (一般)



- **スタイル：**

  最低限：ゲージの基本版です。
  補完：クロム、シェード、その他の機能などの、現実世界のゲージを模倣した、グラフィック的にスタイル化されたゲージ。

フィラー、マーカー、および放射状ゲージ (色範囲)



- **色の範囲：**

  自動：返された値範囲の色とサイズを Splunk が決定します。
  マニュアル：色と値範囲を指定します。

地図

地図視覚エフェクトを作成するには、ダッシュボード内に `geostats` サーチコマンドを使用するパネルを作成します。geostats コマンドの詳細は、『サーチリファレンス』の「geostats」を参照してください。

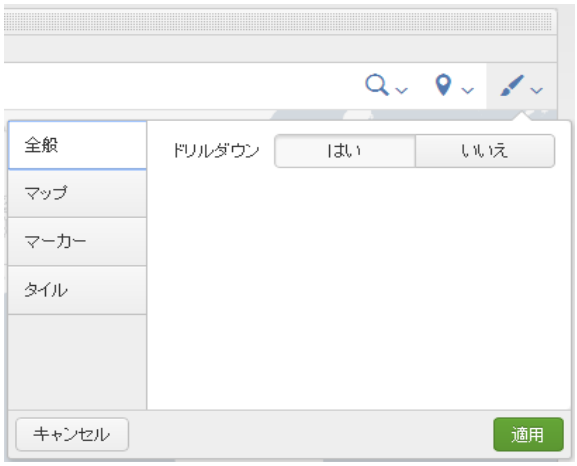
インラインサーチの 2 つの例を以下に示します。最初の例は、Splunk Enterprise に用意されているルックアップテーブルを使用します。2 番目の例は、foursquare から利用できる地図データを参照します。

```
| inputlookup mapdata.csv | geostats latfield=latitude longfield=longitude sum(count) as count
sourcetype=foursquare | geostats latfield=checkin.geolat longfield=checkin.geolong count by checkin.user.gender
```

ビジュアルエディタは、地図の基本プロパティの編集フィールドを提供しています。シンプル XML ソースで地図を編集する際には、その他のプロパティも利用できます。地図の編集に利用できるすべてのプロパティについては、『シンプル XML リファレンス』の「Map エントリ」を参照してください。

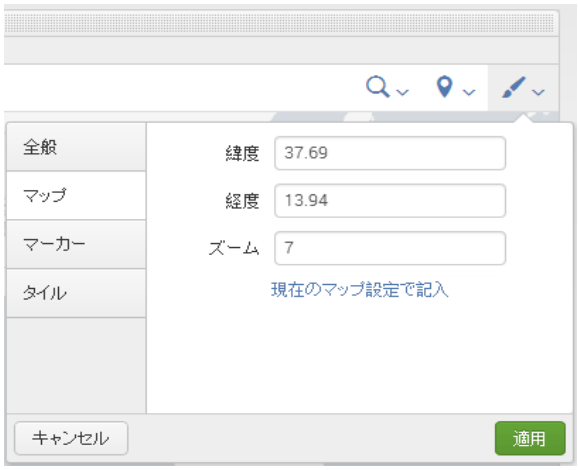
地図のドリルダウン

[全般] プロパティから、ドリルダウン動作を有効/無効にすることができます。



地図の初期設定

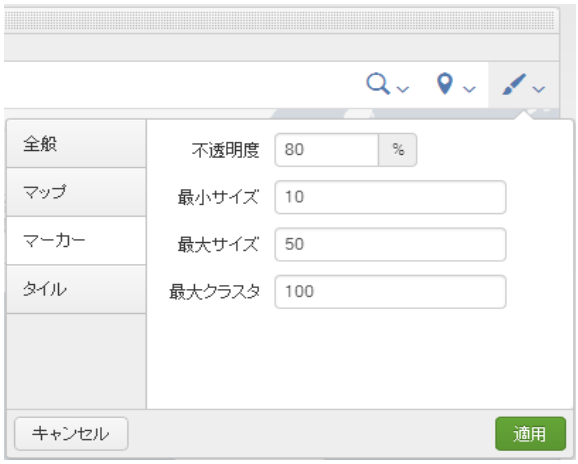
[地図] プロパティから、初期の中心座標と初期のズームレベルを指定します。



- **緯度**：
  初期の中心点の値です。緯度値の範囲は -85～85 で、この範囲外の値は省略されます。
- **経度**：
  初期の中心点の値です。経度値の範囲は -180～180 で、この範囲外の値は省略されます。
- **ズーム**：
  地図の初期ズームレベル。
  Splunk が提供するデフォルトの地図タイルセットでは、7 が最大のズームレベルになります。より大きく拡大したい場合は、別のタイルセットを選択してください。
- **[現在のマップ設定で記入]**
  地図内で、場所とズームレベルをパンすることができます。これらの設定を、地図の初期設定のまま保持する場合は、**[現在のマップ設定で記入]** を選択します。

地図のマーカー設定

[マーカー] プロパティから、マーカーの不透明度とサイズを定義することができます。マーカークラスタの最大数を指定することもできます。

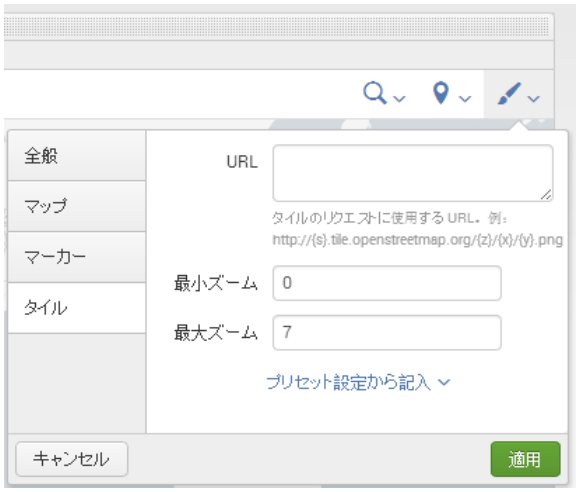


- **不透明度**
  マーカーの不透明度を指定します。値は 0% (透明)～100% (不透明)の範囲で指定します。
- **最小サイズ**
  マーカーの最小サイズ (ピクセル)。
- **最大サイズ**
  マーカーの最大サイズ (ピクセル)。
- **最大クラスタ**
  表示する最大クラスタ数。
  警告：このオプションに大量のクラスタを設定すると、パフォーマンスが大幅に低下する可能性があります。1000 未満の値を指定することをお勧めします。

地図タイル

Splunk には、対話型地図用の Leaflet オープンソース JavaScript ライブラリが用意されています。Leaflet に基づく他のタイルセットを指定することができます。

[タイル] から、地図タイルを要求する場所を指定してください。タイルのズーム制限を指定することもできます。



- **URL**
  地図タイルのリクエストに使用する URL を指定します。空にすると、Splunk タイルが使用されます。
- **最小ズーム**
  タイルセットの最小ズームレベル。
- **最大ズーム**
  タイルセットの最大ズームレベル。
- **プリセット設定から記入**
  利用可能な設定からタイルセットを選択します。このリリースの Splunk は、デフォルトのタイルセットの他に、Open Street Map から利用できるタイルセットのリンクも提供しています。

ダッシュボード PDF の生成

Splunk には、ダッシュボードの PDF を生成、印刷するためのさまざまなオプションが用意されています。

- ダッシュボードの PDF を生成、保存することができます。
- ダッシュボードの PDF をプリンタに送ることができます。
- PDF のメール配信をスケジュールできます。

注意：スケジュール済みレポートの PDF 出力をメールに添付して送信することもできます。設定方法については、『レポートマニュアル』の「レポートのスケジュール」を参照してください。

ダッシュボードの PDF の生成

ダッシュボードから、その PDF をダウンロードまたは印刷することができます。

> **注意：**PDF の生成には制限事項があります。

ダッシュボードの PDF をエクスポートするには

1. ダッシュボードで、[PDF のエクスポート] アイコンをクリックします。
   ブラウザに生成された PDF が表示されます。ブラウザから、PDF の表示、ダウンロード、または印刷を行えます。

   [PDF のエクスポート] および [PDF の印刷] アイコン

ダッシュボードから PDF を印刷するには

1. ダッシュボードで、印刷アイコンをクリックします。
   ブラウザのデフォルトのプリントドライバに、ダッシュボードの PDF を印刷するための設定が表示されます。

リアルタイムサーチと統合 PDF 生成

リアルタイムサーチ の PDF 生成では特別な処理が行われます。

統合 PDF 生成機能を使ってリアルタイムに実行されているサーチ、レポート、またはダッシュボードパネルの PDF を生成する場合、Splunk はそのサーチを履歴サーチに変換します (基本的には、時間範囲から rt を削除する)。そのため、5 分ウィンドウのリアルタイムサーチの場合、PDF には生成時から過去 5 分間の実行結果が表示されます。

ダッシュボードにリアルタイム全時間を対象にしたサーチ結果を表示するパネルがある場合、そのダッシュボードの PDF には全時間を対象にしたサーチ結果が表示されます。

ダッシュボードの PDF 配信のスケジュール

ダッシュボードの PDF 配信をスケジュールすることができます。PDF 配信は、ダッシュボードの [編集] メニューから指定します。[編集] メニューは、ダッシュボードメニューから直接利用することも、[ダッシュボード] ページのダッシュボードのリストから利用することもできます。

> **注意：**PDF 配信機能は、シンプル XML をベースにしたダッシュボードでのみ利用できます。フォームの場合は、それがシンプル XML で作成されている場合でも、PDF 配信をスケジュールすることはできません。HTML に変換したダッシュボードの PDF 配信をスケジュールすることはできません。詳細は、「PDF 生成の制限事項」を参照してください。

メール通知の設定

ダッシュボードをメールに添付した PDF ファイルとして送信する前に、[設定] でメール通知の設定を行う必要があります。この設定は、メールでアラートを通知する設定と同じ設定です。「メール通知の設定」を参照してください。

スケジュール済みダッシュボード配信でのトークンの使用

トークンは、サーチジョブが生成したデータを表すある種の変数です。Splunk Enterprise には、サーチが生成した情報をメールのフィールドに記入するために利用できる、さまざまなトークンが用意されています。スケジュール済み PDF 配信の場合、メールのフィールドに以下のトークンを使用することができます。

- 件名
- メッセージ
- フッター

トークンの値にアクセスするには、以下の構文を使用します。

$<token-name>$

たとえば、スケジュール済み PDF 配信の件名フィールドにダッシュボードを含む App を引用するには、以下のようにトークンを指定します。

app からのサーチ結果

メール通知に利用できるトークン

このセクションは、ダッシュボードのスケジュール済みメール配信に使用できる一般的なトークンを記載していま

す。サーチが生成するデータにアクセスするトークンは、4 つのカテゴリに分けられます。トークンを使用するコンテキストが異なっています。

以下の表には、すべてのカテゴリのトークンを記載しています。PDF 配信のスケジュールには、サーチメタデータとサーバー情報のカテゴリのみが適用されます。

カテゴリ	説明	コンテキスト
サーチメタデータ	サーチに関する情報。	ダッシュボードのスケジュール済み PDF 配信 サーチからのアラートアクション スケジュール済みレポート
サーバー情報	Splunk Enterprise サーバーに関する情報	ダッシュボードのスケジュール済み PDF 配信 サーチからのアラートアクション スケジュール済みレポート
サーチ結果	サーチ結果へのアクセス	サーチからのアラートアクション スケジュール済みレポート
ジョブ情報	サーチジョブ固有のデータ	サーチからのアラートアクション スケジュール済みレポート

このトピックに記載している一般的なトークンの他に、`savedsearches.conf` 設定ファイルには、トークンから値を利用できる属性が記載されています。これらの属性の値を利用するには、各属性をトークン区切り文字「$」で囲んで指定します。

サーチメタデータにアクセスするトークン

サーチに関する情報にアクセスする一般的なトークン。これらのトークンは、ダッシュボードのスケジュール済み PDF 配信に利用できます。

利用できる一般的なトークンを以下の表に示します。

トークン	説明
$action.email.hostname$	メールサーバーのホスト名。
$action.email.priority$	メール配信の優先度。
app	ダッシュボードを含んでいる App 名。
$cron_schedule$	PDF 配信の Cron スケジュール。
$description$	ダッシュボードの説明。
$name$	ダッシュボードの名前。
$next_scheduled_time$	次回のサーチ実行時刻。
$owner$	ダッシュボードの所有者。
$type$	サーチがアラート、レポート、ダッシュボード、またはサーチコマンドからのものかどうかを示します。
$view_link$	ダッシュボードを表示するためのリンク。

サーバーから利用できるトークン

Splunk Enterprise サーバーから利用できる詳細情報を提供する一般的なトークン。これらのトークンは、ダッシュボードのスケジュール済み PDF 配信に利用できます。

利用できる一般的なトークンを以下の表に示します。

トークン	説明
$server.build$	Splunk Enterprise インスタンスのビルド番号。
$server.serverName$	Splunk Enterprise インスタンスのサーバー名。
$server.version$	Splunk Enterprise インスタンスのバージョン番号。

ダッシュボードの PDF 配信のスケジュール

ダッシュボードの PDF 配信をスケジュールするには：

1. 目的のダッシュボードで、**[編集]** > **[PDF 配信のスケジュール]** を選択します。
2. **[PDF 配信のスケジュール]** チェックボックスを選択して、PDF 配信を有効にします。

3. スケジュールの選択
   [Cron スケジュールを実行] を選択した場合は、「cron の例」を参照してください。
4. メール詳細を指定します。
   [件名] および **[メッセージ]** フィールドにトークンを使用することができます。
   - **[To]**、**[CC]** 、および **[BCC]** メール受信者。
     メール受信者のカンマ区切りリストを指定します。
   - **優先度**
     優先度の採用は、メールクライアントによって異なります。
   - **件名**
   - **メッセージ**
5. **[用紙サイズ]** および **[用紙レイアウト]** を選択します。
6. 配信設定を保存するには、**[保存]** をクリックします。

ダッシュボード PDF のメール配信を終了するには

1. 目的のダッシュボードで、**[編集]** > **[PDF 配信のスケジュール]** を選択します。
2. **[PDF 配信のスケジュール]** の選択を解除します。
3. 配信設定を保存するには、**[保存]** をクリックします。

PDF 配信の cron スケジュールの指定

標準の cron 表記を使って、独自の配信スケジュールを定義することができます。**[cron]** を選択すると、cron スケジュールを入力するフィールドが表示されます。

注意：Splunk Enterprise は cron 表記で 5 つのパラメータを使用します (6 つではない)。他の表記で一般的な 6 番目のパラメータ `year` は使用しません。

パラメータを以下に示します。

(* * * * *)

対応

```
minute hour day month day-of-week。
```

以下にいくつかの cron 例を示します。

```
*/5 * * * *       : Every 5 minutes
*/30 * * * *      : Every 30 minutes
0 */12 * * *      : Every 12 hours, on the hour
*/20  * * * 1-5   : Every 20 minutes, Monday through Friday
0 9 1-7 * 1       : First Monday of each month, at 9am.
```

他の PDF 印刷用設定

PDF 印刷に関する、以下の設定を行えます。

- 印刷するテーブルの最大行数
- PDF 生成のタイムアウト設定

- Splunk ロゴの表示場所
- 非ラテンフォント使用の有効化

テーブル内の行数の設定

ダッシュボードパネル内の単純な結果テーブルに対して Splunk が生成するデフォルトの行数は 1000 です。1000 行を超えるテーブルを持つダッシュボードがある場合は、最初の 1000 行のみが PDF に出力されます。必要に応じて、複数のページが作成されます。

PDF として生成されるデフォルトの行数に優先する設定を行うには、`limits.conf` ファイルを使用します。

PDF に出力するテーブル行数の最大値を設定するには：

1. 編集するために、$SPLUNK_HOME/etc/system/local/limits.conf を開きます。
   このファイルが存在していない場合は、ファイルを作成します。
2. `[pdf]` スタンザに以下のプロパティを指定します。

```
[pdf]
max_rows_per_table = <unsigned int>
```

注意：この設定は、Splunk インスタンス内のすべてのテーブルの PDF に適用されます。

PDF 生成のタイムアウト設定

PDF 生成のデフォルトのタイムアウトは 3600 秒です (`limits.conf` に設定されている)。完了までに時間がかかる複雑なサーチの場合、PDF 生成が完了するまでに時間が必要なことがあります。

PDF 生成のタイムアウト値を設定するには：

1. 編集するために、$SPLUNK_HOME/etc/system/local/limits.conf を開きます。
   このファイルが存在していない場合は、ファイルを作成します。
2. PDF の生成までに待機する秒数を指定します。このプロパティは、`[pdf]` スタンザにあります。

```
[pdf]
render_endpoint_timeout = <unsigned int>
```

注意：この設定は、Splunk インスタンス内のすべての PDF 生成タイムアウトに適用されます。

PDF への Splunk ロゴ表示の設定

デフォルトでは、生成された PDF に Splunk ロゴが自動的に表示されます。デフォルトの設定を変更するには、`alert_actions.conf` を使用します。

生成された PDF に Splunk ロゴを表示しないようにするには：

1. 編集するために、$SPLUNK_HOME/etc/system/local/alert_actions.conf を開きます。
   このファイルが存在していない場合は、ファイルを作成します。
2. `[email]` スタンザに以下のプロパティを指定します。

```
[email]
reportIncludeSplunkLogo=0
```

注意：この設定は、Splunk インスタンス内のすべての PDF 生成に適用されます。

非ラテンフォント使用の有効化

Splunk には、ラテンフォントのパッケージの他に、日本語、韓国語、簡体中国語、繁体中国語を処理するための一連の CID フォントが用意されています。

Splunk による CID フォントの読み込みを整序するためには、`alert_actions.conf` 内の `reportCIDFontList` パラメータを変更します。フォントはスペースで区切ったリスト形式で指定します。特定の文字コードの記号を複数のフォントが提供している場合は、リスト内で最初に指定されたフォントの記号が使用されます。

`reportCIDFontList` パラメータは、`[email]` スタンザ内にあります。ここから、フォントの使用方法を変更してください。

```
$SPLUNK_HOME/etc/system/local/alert_actions.conf
```

Splunk がデフォルトでサポートしている CID フォントを以下に示します。

```
gb cns jp kor
```

これらはそれぞれ簡体中国語、繁体中国語、日本語、および韓国語を表しています。

CID フォントのロードをスキップするには、ローカル版の `alert_actions.conf` で、`reportCIDFontList` の値を空にしてください。

PDF で別の TrueType 非ラテンフォント (Cyrillic や Greek など)を使用する場合は、Splunk 管理者に `$SPLUNK_HOME/share/splunk/fonts` への Unicode フォントの追加を依頼してください。`fonts` ディレクトリが存在していない場合は、ディレクトリを作成します。

注意：複数のフォントがインストールされている場合、それらはアルファベット順にソートされます。そのため、Cyrillic と Greek をインストールした場合、`$SPLUNK_HOME/share/splunk/fonts` ファイルで Greek が先に来るようにファイル名を変更しない限り、常に Cyrillic が選択されます。

PDF 生成の制限事項

Splunk の統合 PDF 生成機能にはいくつかの制限事項があります。

- アドバンスト XML または HTML を使って作成されたダッシュボードの PDF は生成できません。Splunk では、シンプル XML で作成されたダッシュボードの PDF 生成のみがサポートされています。
- フォームの PDF は生成できません。
- PDF 生成でヒートマップは無視されます。ダッシュボードの残りの部分は出力されますが、ヒートマップが提供するシェーディングは反映されません。
- PDF 生成で、JSChart ライブラリがサポートしていない、グラフのカスタマイズは無視されます。生成される PDF には、JSChart が描画したパネルが表示されますが、サポートされていないカスタマイズ機能は適用されません。

ダッシュボードの HTML への変換

シンプル XML では利用できない機能を持つダッシュボードを利用したい場合は、ダッシュボードを HTML にエクスポートすることができます。生成される HTML は、SplunkJS スタックをベースにしています。次に HTML と JavaScript を編集して、独自の機能を実装します。

HTML ダッシュボードの編集用リソース

HTML と JavaScript の編集は、このマニュアルの対象外です。SplunkJS スタックをベースにしたダッシュボードの編集については、Splunk Developer Portal の以下の記事を参照してください。

- Web Framework Concepts (Web フレームワークの概念)
- SplunkJS Stack
- HTML Dashboards

シンプル XML ダッシュボードの HTML への変換

デフォルトで、Splunk はシンプル XML をベースにしたダッシュボードを作成します。ダッシュボードを、Splunk Web フレームワークの SplunkJS スタックコンポーネントに由来する HTML ベースに変換/エクスポートすることができます。

Splunk のビジュアルエディタを使って、生成された HTML ソースコードを編集することはできません。代わりに Splunk のソースエディタまたはサードパーティ製のエディタを使用してください。

> **注意：**この HTML ダッシュボードに対して、統合 PDF 生成を利用することはできません。

ダッシュボードのソースコードを HTML に変換するには：

1. シンプル XML のダッシュボードで、**[編集]** > **[HTML に変換]** を選択します。
2. 以下の事項を指定して、**[ダッシュボードの変換]** をクリックします。
   - **[新規作成]** または **[現状と置換]**を選択します。
     現在のダッシュボードと置換すると、その基盤となるシンプル XML は利用できなくなります。新しいダッシュボードを作成して、元のシンプル XML コードは残しておくことをお勧めします。
   - タイトル
   - ID
   - 説明
   - 権限
     ダッシュボードを変換したら、その権限を編集することができます。

HTML ダッシュボードの編集

SplunkJS スタック由来の HTML に変換したダッシュボードを編集する場合、一般的には HTML を編集するだけではなく、その HTML コードが参照している CSS や JavaScript へのアクセスも必要になります。

Splunk には、ダッシュボードの HTML ソースコードの編集に利用できる XML エディタが用意されています。ただし、関連する CSS および JavaScript ファイルに Splunk サーバーがアクセス、編集できるように、ローカルファイルシステムへのアクセス権も必要となります。サードパーティのエディタを使ってソース HTML コードを編集する場合も、ローカルファイルシステムへのアクセスが必要です。

ソースファイルへのアクセスの詳細は、「ダッシュボードとフォームのソースファイル」を参照してください。

Splunk Developer Portal の HTML ダッシュボードの編集に関するドキュメントへのリンクについては、「HTML ダッシュボードの編集用リソース」を参照してください。

ダッシュボード編集の主要ファイル

ダッシュボードを HTML に変換した場合の、ソースファイルの場所については、「ダッシュボードとフォームのソースファイル」を参照してください。

HTML ソースファイルの編集にサードパーティ製のエディタを使用する場合、編集後のソースファイルを反映したダッシュボードを表示するために、忘れずに Splunk インスタンスを更新し、再表示してください。例：

http://localhost:8000/en-US/debug/refresh

CSS、JavaScript、または静的 HTML ファイルを更新した場合は、編集したダッシュボードを表示するブラウザページの更新しか必要ありません。

シンプル XML を使ったダッシュボードの作成

シンプル XML の編集について

デフォルトで、Splunk Enterprise のダッシュボードはその基盤となるコードにシンプル XML を使用しています。Splunk Enterprise の対話型エディタを利用すれば、シンプル XML を記述/編集せずにダッシュボードを作成、編集することができます。ただし、対話型エディタでは一部の高度なダッシュボード機能を利用できません。これらの機能を利用するには、基盤となるシンプル XML コードを編集します。

この章では、シンプル XML 編集の基本的な事項について説明していきます。シンプル XML のダッシュボードやフォームで利用できる機能の詳細な例を取り上げています。

ダッシュボードとフォームの構造の詳細は、「ダッシュボードとフォームの構造」を参照してください。

シンプル XML 構文の詳細は、「シンプル XML リファレンス」を参照してください。

XML エディタ

ダッシュボード/フォームのシンプル XML ファイルを編集するには、2 種類の方法が存在しています。

- Splunk Enterprise の XML ソースエディタを使ってシンプル XML を編集する。
  Splunk XML エディタの使用方法は、「Splunk Web を使ったダッシュボードの作成と編集」を参照してください。「ソース XML の編集」に、エディタへのアクセスと使用方法が記載されています。
- 任意のサードパーティ製 XML エディタを使って、シンプル XML ソースファイルを編集します。
  サードパーティ製エディタを使用する場合に必要な Splunk 環境に関する情報については、下記を参照してください。

サードパーティ製 XML エディタ

サードパーティ製のエディタを使って XML を編集する場合、以下の事項を理解しておく必要があります。

- XML ファイルと関連 CSS および JavaScript ファイルにアクセスできるように、Splunk Enterprise インストールに対する書き込みアクセス権が必要です。書き込みアクセス権がない場合は、Splunk Enterprise の管理者に問い合わせてください。
- ソースファイルのファイルシステム権限を確認してください。
  ファイルをコピーした後に、ファイルを読み込み/書き込みできる必要があります。これは、Splunk Web のダッシュボードのユーザーのアクセス権の設定とは異なります。
- Splunk Enterprise インスタンスを更新します。
  ダッシュボード XML ファイルを変更したら、それを反映するために Splunk Web で Splunk Enterprise インスタンスを更新する必要があります。Splunk Enterprise のソースエディタを使用する場合は、この操作は不要です。Splunk Enterprise インスタンスを更新するには、以下の URL を使用します。インスタンスを更新したら、ダッシュボードを再ロードします。たとえば、ローカル版の Splunk Enterprise のデフォルトは次のようになります：

  http://localhost:8000/debug/refresh

XML ファイル内の特殊文字

XML ファイル内で、一部の文字は特別な意味を持っており、その文字通りの意味では使用できません。CDATA タグ内のテキストは、以下のように指定して折り返すことができます。XML パーサーは、CDATA タグ内のテキストを処理しません。

```
<![CDATA[
 <code>"Text within a CDATA tag"</code>
]]>
```

また、HTML エンティティを使ってこれらの文字をエスケープ処理することができます。

文字	HTML エンティティ
'	&apos;
<	&lt;
>	&gt;
&	&amp;

ダッシュボードとフォームのソースファイル

ダッシュボードのソースファイルには、シンプル XML ファイル、JavaScript ファイル、および CSS が含まれます。ソースファイルには、参照先としてインポートした静的 HTML ファイルと画像ファイルも含まれます。

ソースファイルが Splunk Enterprise インスタンス内の特定の場所に存在するものとして処理されます (後述)。

ビューとパネルのソース・ファイル

App の **views** ディレクトリには、以下のソース・ファイルが含まれています。

- シンプル XML で作成されたビュー。
- 従来のアドバンスト XML で作成されたビュー。
- ダッシュボード内で参照を使って利用できるパネル・ファイル。「参照によるパネルの作成と追加」を参照してください。

ソース・ファイルの場所は、ダッシュボードの権限が **[App 内で共有]** か、または **[プライベート]** かによって異なります。

シンプル XML、アドバンスト XML、およびパネルのソース・ファイルは、以下の場所に保管します。

共有している場合
$SPLUNK_HOME/etc/apps/<app>/local/data/ui/**views**/<ファイル名>

プライベートの場合
$SPLUNK_HOME/etc/users/<ユーザー>/<app>/local/data/ui/**views**/<ファイル名>

App の **html** ディレクトリには、HTML に変換されたビューのソースファイルが含まれています。ソース・ファイルの場所は、ダッシュボードの権限が **[App 内で共有]** か、または **[プライベート]** かによって異なります。

HTML ソースファイルは以下の場所に保管します。

共有している場合
$SPLUNK_HOME/etc/apps/<App>/local/data/ui/**html**/<ダッシュボードファイル名>

プライベートの場合
$SPLUNK_HOME/etc/users/<ユーザー>/<App>/local/data/ui/**html**/<ダッシュボードファイル名>

default または local

ソースファイルは、default または local ディレクトリを使用するパスに保管できます。ただし一般的には、local ディレクトリを含むパスに、ダッシュボードのソースファイルを保管します。

local ディレクトリが default ディレクトリよりも優先されます。つまり、まず最初に local ディレクトリのリソースが参照されます。local ディレクトリに存在しているリソースはすべて、default ディレクトリに存在している同じリソースよりも優先されます。

default ディレクトリにソースファイルを配置すると、更新時に変更内容が失われてしまいます。Splunk Enterprise の更新時、または Splunk Enterprise インスタンス内の App の更新時に、default ディレクトリの内容は上書きされてしまいます。ただし、local ディレクトリの内容はそのまま残されます。更新後に失われることがないように、ダッシュボードのシンプル XML ソースファイルは、local ディレクトリに保管するようにしてください。前のシンプル XML ソースファイルの例は、local ディレクトリを使用しています。

ダッシュボードエディタは、`local` のパスに書き込みます。

ファイルの優先順位については、「設定ファイルの優先順位」を参照してください。

CSS、JavaScript、および他の静的ファイル

ダッシュボードには、CSS や JavaScript ファイルだけでなく、画像ファイルや静的 HTML ファイルをインポートすることもできます。これらのファイルは以下の場所にあります。ファイルをサブディレクトリに配置することはできません。

$SPLUNK_HOME/etc/apps/<App 名>/appserver/static/

デフォルトで、このディレクトリには、以下の 2 つのファイルが含まれています。

- dashboard.css
- dashhboard.js

この場所にあるデフォルトファイルを編集したり、他の CSS および JavaScript ファイルを追加したりできます。また、ダッシュボードに参照させる他の HTML ファイルを追加することもできます。

JavaScript および CSS ファイルのインポート

<dashboard> または <form> に `script` および `stylesheet` 属性を使って、App のデフォルトの場所から JavaScript または CSS ファイルをインポートできます。他の App からの スクリプトまたは CSS ファイルを参照することもできます。

例：

同じ App からのファイルのインポート

```
<dashboard script="myScript.js" stylesheet="myStyles.css">
. . .
</dashboard>
```

他のApp「myApp」からのファイルのインポート

```
<dashboard script="myApp:myScript.js" stylesheet="myApp:myStyles.css">
. . .
</dashboard>
```

ダッシュボードとフォーム

ダッシュボード、フォーム、およびパネル・ファイルのシンプル XML ソース・コードを編集する前に、ダッシュボードの基本的なレイアウトとそれを定義する XML エレメントについて理解しておく必要があります。基本的な構造については、「ダッシュボードとフォームの構造」を参照してください。

「シンプル XML リファレンス」と「グラフ設定リファレンス」には、ダッシュボードとフォームの作成に利用できる、すべてのシンプル XML エレメントおよびオプションの詳細が記載されています。コード記述の詳細は、これらのリファレンスを参照してください。

ダッシュボードとフォームの基盤となるサーチ

Splunk Enterprise のサーチは、ダッシュボード、フォーム、そしてそれに含まれているデータの視覚エフェクトの基盤となっています。ここでは、利用できるサーチの種類、およびシンプル XML を使ったダッシュボードやパネルへのサーチの追加方法の概要を説明していきます。

Splunk Enterprise のサーチ言語を初めて使用する場合は、『サーチ・マニュアル』の「サーチについて」を参照してください。ご自分のデータの重要な側面を探し出し、ユーザーの目標達成を支援するようなサーチを作成してください。『サーチリファレンスマニュアル』には、より良いサーチの作成方法、サーチコマンドリスト、および Splunk の各サーチコマンドの詳細情報など、サーチに関するさまざまな情報が記載されています。

ダッシュボード内のサーチの概要

ダッシュボードのコンテンツを生成するサーチにアクセスするには、さまざまな方法があります。

インラインサーチ

インライン・サーチは、ダッシュボードまたは視覚エフェクト内に作成した**サーチ**です。

ダッシュボード内でグローバルなインライン・サーチ、またはダッシュボード内の各視覚エフェクト用のインライン・サーチを指定できます。ダッシュボードでグローバルなサーチには、視覚エフェクト内で後処理サーチが必要になります。後処理サーチは、グローバル・サーチから返されたデータをさらに調整します。

「インライン・サーチの例」を参照してください。

レポートとして保存されたサーチ

サーチを**レポート**として保存して、ダッシュボードからそのレポートを参照することで、ダッシュボードでサーチを利用することができます。詳細は、『レポート・マニュアル』の「レポートの作成と編集」を参照してください。

「レポートからのサーチの参照」の例を参照してください。

ピボットを使ったサーチの生成

Splunk Enterprise のピボット・ツールを使って、サーチを**ピボット**として生成できます。ピボットは、ダッシュボードにエクスポートすることができます。詳細は、『ピボット・マニュアル』を参照してください。「ピボットエディタを使ったピボットテーブルの設計」には、ピボットの作成とサーチとしてのエクスポート方法の詳細が記載されています。

フォームへの入力用サーチ

サーチを使って、ラジオ・ボタン、ドロップダウン・リスト、チェックボックスなどのフォーム入力用の選択項目を、動的に設定できます。

「入力用選択項目の設定」の例を参照してください。

サーチでのトークンの使用

サーチに、トークンを利用することができます。トークンは、サーチのフィールドとその値を参照する、ある種の変数です。サーチ・コマンドにトークンを定義するには、フィールドを `$...$` 文字で囲みます。以下のコード・スニペットでは、前にトークンが `$series_tok$` で定義されています。

```
index=_internal source=*metrics.log group="per_sourcetype_thruput" series=$series_tok$ | table sourcetype eps, kb,
kbps
```

フォーム内のトークンを使ってユーザー入力を受け付けて、ダッシュボードにラベルとタイトルを表示します。

「基本的なフォームの例」は、フォーム内でのトークンの使用方法について説明しています。「ダッシュボードでのトークンの使用」も参照してください。

ダッシュボード内のサーチのシンプル XML エレメント

シンプル XML でサーチを定義するには、<search> エレメントとその子エレメントを使用します。<query> エレメントには、実際のサーチ文字列を指定します。<earliest> および <latest> エレメントには、サーチの境界を指定します。

search エレメントは以下の場合に使用します。

- 視覚エフェクトのデータを生成するサーチ。
- ダッシュボードまたはフォームのグローバル・サーチ。
  グローバル・サーチを参照するためには、視覚エフェクトで後処理サーチを使用します。
- 視覚エフェクトの後処理サーチ。
  後処理サーチは、グローバル・サーチから返されたデータを変更します。
- ラジオ入力やドロップダウン入力などの、入力用のラベルと値を返すサーチ。

シンプル XML コードでのサーチの作成方法については、『シンプル XML リファレンス』の「search エレメント」を参照してください。

シンプル XML のサーチ例

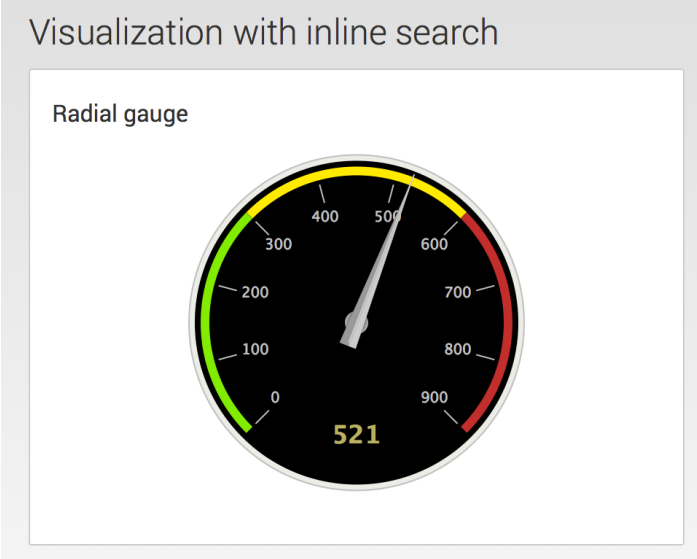
ここでは、以下の場合の <search> エレメントの使用例を取り上げていきます。

- インライン・サーチ。
- レポートからのサーチの参照。
- 入力用選択項目の設定。
- グローバル・サーチを参照する後処理サーチ。

インライン・サーチの例

この例のサーチは、視覚エフェクトのデータを生成します。

- <query>
  サーチ文字列を指定します。
- <earliest> <latest>
  サーチの境界を定義します。




```
<dashboard>
  <label>Visualization with inline search</label>
  <description></description>
  <row>
    <panel>
      <chart>
        <title>Radial gauge</title>

        <search>
          <!-- Inline search query -->
          <query>
            index=_internal source="*splunkd.log"
```

```
          ( log_level=ERROR OR log_level=WARN*
          OR log_level=FATAL OR log_level=CRITICAL )
          | stats count as log_events
           | rangemap field=log_events low=1-100 elevated=101-300 default=severe
        </query>

        <!-- search bounds -->
        <earliest>-7d@h</earliest>
        <latest>now</latest>
      </search>

      <option name="charting.chart">radialGauge</option>
      <option name="charting.chart.rangeValues">[0,300,600,900]</option>
    </chart>
  </panel>
 </row>
</dashboard>
```

レポートからのサーチの参照

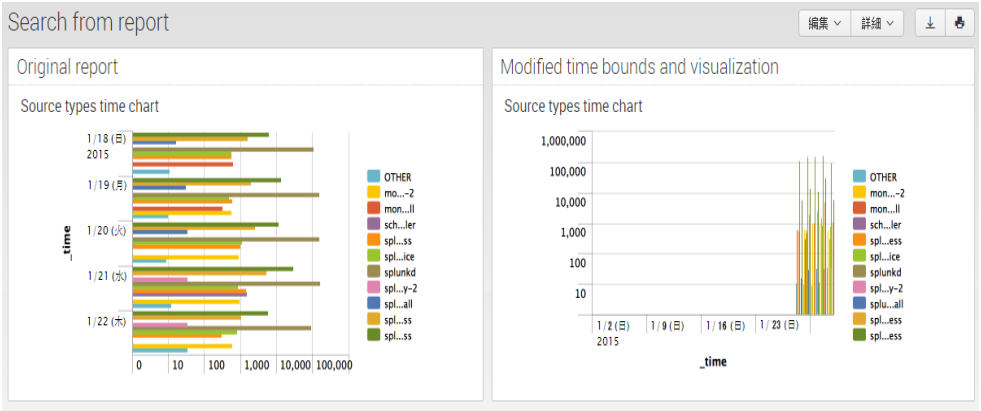
この例のサーチは、レポートを参照します。

ダッシュボードからサーチを変更することはできませんが、サーチ結果の時間境界と視覚エフェクトを変更することは可能です。レポート内のサーチが変更されると、そのレポートに基づく視覚エフェクトにも変更が反映されます。

この例のレポートは視覚エフェクトの横棒グラフを使用して、過去 7 日間の結果を表示します。左側のパネルは、レポートからのサーチを表示します。右側のパネルは、同じレポートからのサーチを使用しますが、時間境界と視覚エフェクトが変更されています。

<search> エレメントのコード：

- <search ref ="[name]">
  レポートを参照します。
- <earliest> <latest>
  時間境界を変更します。




```
<dashboard>
  <label>Search from report</label>
  <row>
    <panel>
      <title>Original report</title>
      <chart>
        <title>Source types time chart</title>

        <search ref="Source types time chart" />

      </chart>
    </panel>
    <panel>
      <title>Modified time bounds and visualization</title>
      <chart>
        <title>Source types time chart</title>

        <search ref="Source types time chart">

          <!-- Modify time bounds -->
```

```
        <earliest>-30d@d</earliest>
        <latest>now</latest>

      </search>

      <!-- Change visualization -->
      <option name="charting.chart">column</option>

    </chart>
  </panel>
</row>
</dashboard>
```

フォーム入力の選択項目の設定

以下のフォーム入力の選択項目を動的に設定するには、search エレメントを使用します。

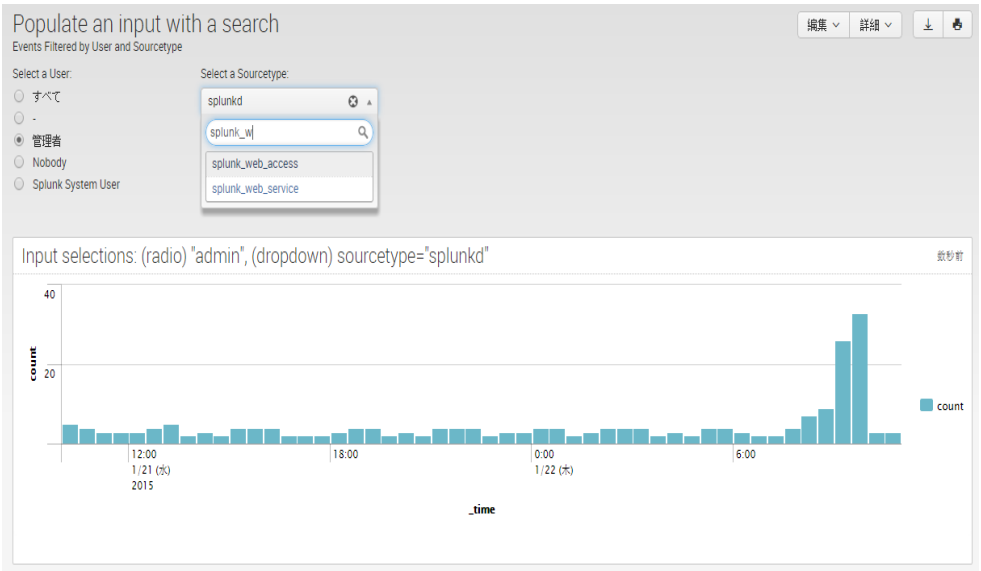
- チェックボックス
- ドロップダウン・リスト
- 複数選択入力
- ラジオ・ボタン

警告：設定用サーチにリアルタイム・サーチは使用しないでください。リアルタイム・サーチを使用すると、入力用の選択項目が正しく更新されません。

この例のサーチは、選択項目の静的定義と動的定義を比較しています。ドロップダウン・リストは、設定用サーチを使って選択項目を定義しています。

- 設定用 <search>
  選択項目のラベルと値に使用するフィールドを返します。

- <fieldForLabel> <fieldForValue>
  <input> エレメントの子エレメント。これらのエレメントは、ドロップダウンの選択項目を設定するために使用するフィールドを指定します。




```
<form>
  <label>Populate an input with a search</label>
  <description>Events Filtered by User and Sourcetype</description>
  <!-- Do not need a Search Button. Inputs search when changed -->

  <fieldset autoRun="true" submitButton="false">

    <!-- Static definition of choices -->
    <input type="radio" token="username_tok" searchWhenChanged="true">
      <label>Select a User:</label>

      <!-- Define the default value -->
      <default>All</default>

      <!-- Hard-code the choices -->
      <choice value="*">All</choice>
```

```
      <choice value="-">-</choice>
      <choice value="admin">Admin</choice>
      <choice value="nobody">Nobody</choice>
      <choice value="splunk-system-user">Splunk System User</choice>
    </input>

    <!-- Dynamic definition of choices -->
    <input type="dropdown" token="sourcetype_tok" searchWhenChanged="true">
      <label>Select a Sourcetype:</label>
      <prefix>sourcetype="</prefix>
      <suffix>"</suffix>

      <!-- Define the default value -->
      <default>splunkd</default>

      <!-- Hard-code the choice for "All" -->
      <choice value="*">All</choice>

      <!-- Define the other choices with a populating search -->
      <search>
        <query>
          index=_internal | stats count by sourcetype
        </query>
      </search>
      <fieldForLabel>sourcetype</fieldForLabel>
      <fieldForValue>sourcetype</fieldForValue>
    </input>

  </fieldset>
  <row>
    <panel>
      <!-- Use tokens from the <input> elements in the panel title -->
      <title>
        Input selections: (radio) "$username_tok$", (dropdown) $sourcetype_tok$
      </title>

      <chart>

        <!-- search for the visualization, references the input tokens-->
        <search>
          <query>
            index=_internal user=$username_tok$ $sourcetype_tok$ | timechart count
          </query>
          <earliest>-24h@h</earliest>
          <latest>now</latest>
        </search>
       </chart>

    </panel>
  </row>
</form>
```

後処理サーチ

類似の複数のサーチを実行するダッシュボードを作成することもあります。このような場合、ダッシュボードのベース・サーチを作成すれば、サーチ・リソースを節約できます。ダッシュボード内の各パネルで後処理サーチを使用して、ベース・サーチの結果をさらに調整していきます。ベース・サーチとして、ダッシュボードのグローバル・サーチを利用することも、ダッシュボード内の他の任意のサーチを利用することも可能です。

一般的にグローバルサーチは、**変換サーチ**になります。変換サーチは**変換コマンド**を使って、サーチから返されたイベントデータを統計データテーブルに変換します。変換コマンドとサーチの詳細は、『サーチ・マニュアル』を参照してください。

ベース・サーチが単一のインデックス上にある場合、リソースを節約するために後処理サーチが効果的です。サーチヘッドで複数のインデクサーを利用しているような環境では、リソースの節約に後処理サーチがさほど効果を発揮しない場合もあります。この場合、ダッシュボード内で同じサーチを複数回使用する方が効果的なこともあります。

以下のようなことに起因する、後処理サーチの制限事項に注意してください。

- 10,000 件を超えるイベントを返すベース・サーチ。
- 完了までに 30 秒を超えるサーチ操作による、Splunk Web タイムアウト。

これらの制限事項の詳細、および後処理サーチ使用時の注意については、「後処理の制限事項」を参照してください。「後処理の例」には、後処理サーチ作成のガイダンスが記載されています。

後処理の制限事項

後処理サーチには制限があります。

- ベース・サーチが非変換サーチの場合、返された最初の 10,000 件のイベントのみが保持されます。後処理サーチは、この 10,000 件のイベント制限を超えたイベントを処理せず、単純に無視します。そのため後処理サーチのデータは不完全になります。この制限を回避するには、ベース・サーチに変換サーチを使用します。
- 後処理操作に時間がかかると、Splunk Web クライアントのタイムアウト値 30 秒 (変更不可) を超えてしまう可能性があります。そのため、`splunkd` デーモン/サービスの応答がないことによるタイムアウトが発生する可能性があります。一般的にこのような状況は、ベースサーチに非変換サーチを使用した場合に発生します。

raw イベントを返すベース・サーチの回避

raw イベントを返す非変換サーチを使って、次に後処理サーチで変換コマンドを使用するのは妥当に思えるかもしれません。ただし、ベース・サーチが 10,000 件のイベント数制限を超えるデータを返す可能性があります。そうなると、後処理サーチには不完全なデータ・セットが渡されてしまい、ダッシュボードに誤った結果が表示されてしまいます。

10,000 件のイベント数制限を回避するには、ベース・サーチで変換コマンドを使用します。『サーチ・マニュアル』の「変換コマンドとサーチについて」を参照してください。

ベース・サーチで名前が付けられていないフィールドを後処理サーチで参照しない

あるフィールドを後処理サーチでのみ参照するのは妥当なように思えますが、ベース・サーチ内でフィールドのデータを分離しておくことをお勧めします。そうしないと、後処理サーチでのみ参照するフィールドがすべての行で NULL になり、0 件の結果が返されることになります。

この問題を回避するには、ベース・サーチで変換コマンドを使用します。

ベースサーチで大量の行を返さない

データ・キューブから後処理サーチに大量のサーチ結果を渡すと、問題が発生することがあります。

サーバーのタイムアウト

後処理操作に時間がかかりすぎると、パフォーマンス上の問題が発生し、タイムアウトが発生する可能性があります。このような場合は、以下の事項を検討してください。

- ベースサーチが返す結果数とフィールド数。
- これらの結果に対する後処理操作の複雑性。

不完全なデータ

ベース・サーチが 10,000 件のイベント数制限を超えるデータを返す非変換サーチの場合、下流のパネルには前述のように不完全なデータセットが渡されます。ベースサーチでは 10,000 件のイベント制限を回避するように、変換コマンドを使ったデータキューブを構築してください。

後処理の例

後処理は、ベース・サーチで変換コマンドを使用して、結果のフォーマットを変更するような場合に効果を発揮します。

そのためには、特定の基準に従ってテーブルやグラフを作成します。たとえば、同じデータセットから別の視覚エフェクトやレポートを作成することができます。また、元のレポートからさらなる集計を行うことも可能です。

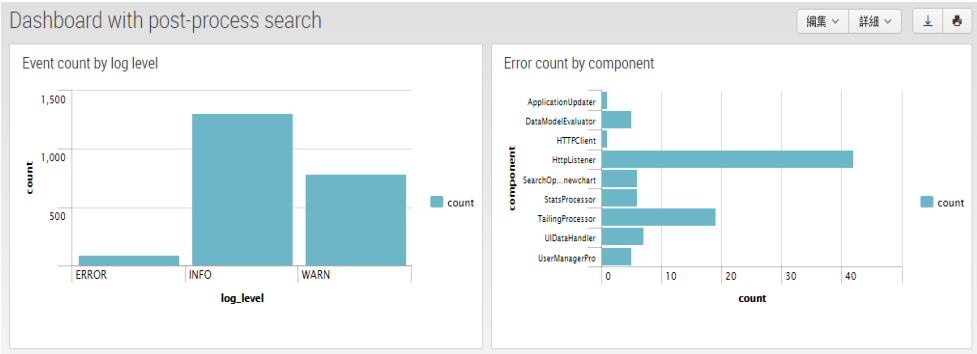
基本的な後処理の例

この例では、ベース・サーチで変換コマンドを使用して、後処理で結果をそれぞれ別の方法で処理します。

ベース・サーチ (データ・キューブ・サーチ)
index=_internal source=*splunkd.log | stats count by component, log_level

後処理 1 (log_level によるイベントのカウント)
| stats sum(count) AS count by log_level

後処理 2 (component によるエラーのカウント)
| search log_level=error | stats sum(count) AS count by component




```
<dashboard>
  <label>Dashboard with post-process search</label>

  <!-- Base search cannot pass more than 10,000 events to post-process searches-->
  <!-- Example uses stats transforming command -->
  <!-- This limits events passed to post-process search -->
  <search id="baseSearch">
    <query>
      index=_internal source=*splunkd.log | stats count by component, log_level
    </query>
  </search>

 <row>
    <panel>
      <chart>
        <title>Event count by log level</title>

        <!-- post-process search -->
        <search base="baseSearch">
          <query>
            stats sum(count) AS count by log_level
          </query>
        </search>

      </chart>
    </panel>
    <panel>
      <chart>
        <title>Error count by component</title>

        <!-- post-process search -->
        <search base="baseSearch">
          <query>
            search log_level=error | stats sum(count) AS count by component
          </query>
        </search>

        <option name="charting.chart">bar</option>
      </chart>
    </panel>
  </row>
</dashboard>
```

複雑な後処理の例

パーセンタイル、標準偏差、平均などの統計処理を含む複雑なベース・サーチの場合、ベース・サーチでサマリー・インデックス・コマンドを使用することをお勧めします。そうすることによって、後処理サーチの作成が容易になります。サマリー・インデックス・コマンドの例を以下に示します。

- sistats
- sitimechart
- sitop
- sichart
- sirare

サマリー・インデックス・コマンドにより、後処理サーチを柔軟に作成することができます。「サマリー・インデックスを使ったレポート効率の向上」および「変換コマンドとサーチについて」を参照してください。

ベース・サーチ (データ・キューブ・サーチ)
index=_internal | eval event_size=len(_raw)

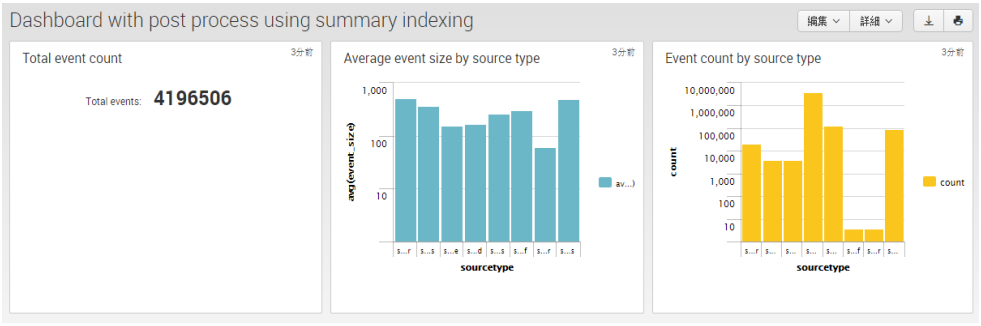
| sistats count min(event_size) avg(event_size) max(event_size)
by source sourcetype

後処理 1
| stats count

後処理 2
| stats avg(event_size) by sourcetype

後処理 3
| stats count by sourcetype

ベース・サーチは _internal インデックスのソースおよびソースタイプ別にイベント・サイズ (min、avg、max) をレポートします。sistats count と各種 group-by 句を使用します。分散サーチ環境では、以下を含めないと map-reduce の利点が失われてしまいます。

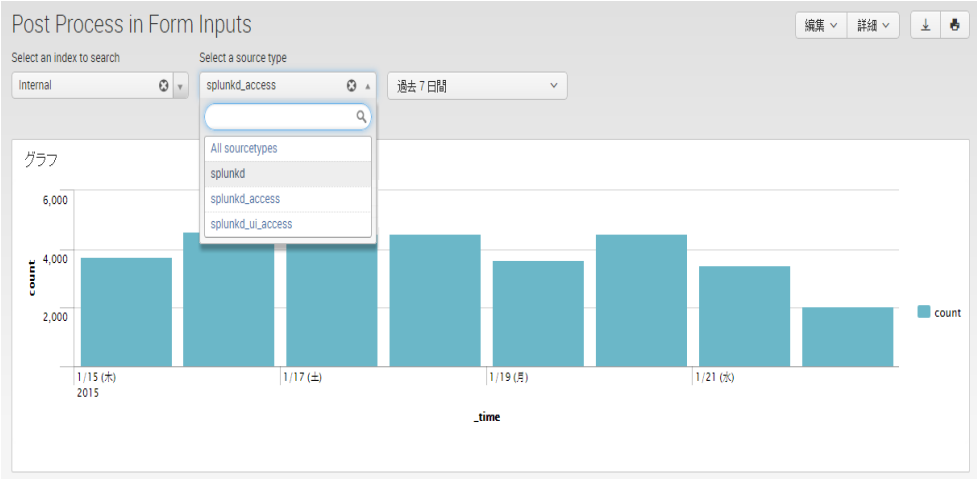



```
<dashboard>
  <label>Dashboard with post process using summary indexing</label>

  <!-- Base search cannot pass more than 10,000 events to post-process searches-->
  <!-- Use summary indexing transforming command -->
  <search id="baseSearch">
    <query>
      index=_internal | eval event_size=len(_raw)
      | sistats count min(event_size) avg(event_size) max(event_size)
        by source sourcetype
    </query>
  </search>

  <row>
    <panel>
      <single>
        <title>Total event count</title>

        <!-- post-process search -->
        <search base="baseSearch">
          <query>stats count</query>
        </search>

        <option name="beforeLabel">Total events: </option>
      </single>
    </panel>
    <panel>
      <chart>
        <title>Average event size by source type</title>

        <!-- post-process search -->
        <search base="baseSearch">
          <query>stats avg(event_size) by sourcetype</query>
        </search>

        <option name="charting.axisY.scale">log</option>
      </chart>
    </panel>
    <panel>
      <chart>
        <title>Event count by source type</title>

        <!-- post-process search -->
```

```
        <search base="baseSearch">
          <query>stats count by sourcetype</query>
        </search>

        <option name="charting.axisY.scale">log</option>
      </chart>
    </panel>
  </row>
</dashboard>
```

入力用後処理サーチを持つフォーム

後処理サーチを使って、フォームの入力を動的に設定することができます。2 つの入力を持つフォームの例を以下に示します。サーチ対象インデックスを選択するドロップダウン・リストには、選択項目が静的に定義されています。ソースタイプを選択するドロップダウン・リストには、デフォルトの選択項目が静的に定義されていますが、他の選択項目は後処理サーチを使って動的に定義されます。



ソースタイプ・ドロップダウン設定用のベース・サーチ
index=_internal | stats count by sourcetype

ドロップダウン入力の後処理
| search sourcetype=splunkd*

```
<form>
  <label>Post Process in Form Inputs</label>

  <!-- Global search for post process by dropdown input -->
  <!-- Search uses stats command to limit results to less than 10,000 limit -->
  <search id="searchInput">
    <query>index=_internal | stats count by sourcetype</query>
    <earliest>-60min</earliest>
    <latest>now</latest>
  </search>

  <fieldset submitButton="false">

    <!-- Select an index from two static choices -->
    <input type="dropdown" token="index_tok" searchWhenChanged="true">
      <label>Select an index to search</label>
      <choice value="_internal">Internal</choice>
      <choice value="*">All public indexes</choice>
      <default>_internal</default>
    </input>

    <!-- Dynamically populate choices -->
    <input type="dropdown" token="sourcetype_tok" searchWhenChanged="true">
      <label>Select a source type</label>

      <!-- default choice is all sourcetypes -->
      <choice value="*">All sourcetypes</choice>
      <default>*</default>
```

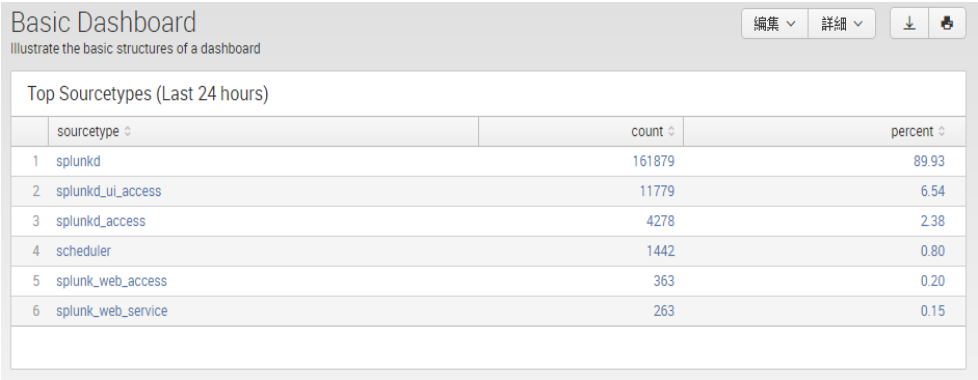
```
      <!-- Post-process search to dynamically populate choices -->
      <search base="searchInput">
        <query>search sourcetype=splunkd*</query>
      </search>
      <fieldForLabel>sourcetype</fieldForLabel>
      <fieldForValue>sourcetype</fieldForValue>

    </input>
    <input type="time" token="time_tok" searchWhenChanged="true">
      <label></label>
      <default>
        <earliest>-24h@h</earliest>
        <latest>now</latest>
      </default>
    </input>
  </fieldset>
  <row>
    <panel>
      <chart>
        <title>Chart</title>
        <search>
          <query>
            index=$index_tok$ sourcetype=$sourcetype_tok$ | timechart count
          </query>
          <earliest>$time_tok.earliest$</earliest>
          <latest>$time_tok.latest$</latest>
        </search>
      </chart>
    </panel>
  </row>
</form>
```

ダッシュボードの例

ここでは、ダッシュボードを構成するシンプル XML ソースコードについて説明しています。シンプル XML ソースコードを理解したら、ダッシュボードをさらにカスタマイズできるようになります。

基本のダッシュボード

この基本のダッシュボードは、基本的なシンプル XML エレメントを使用しています。コード内のコメントには、その説明が記載されています。



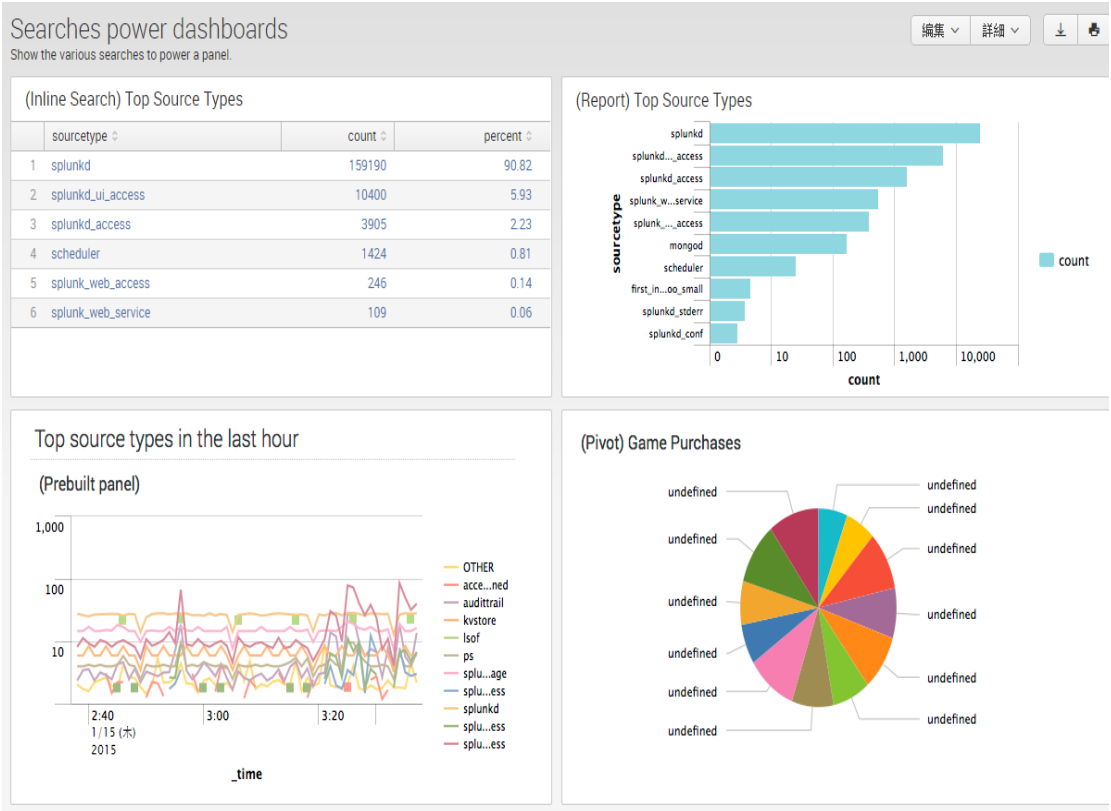

```
<dashboard>
  <!-- A title for the dashboard -->
  <label>Basic Dashboard</label>

  <!-- Provide a description -->
  <description>Illustrate the basic structures of a dashboard</description>

  <!-- Place panels within rows -->
  <row>
```

```
    <!-- This basic dashboard has only a single panel -->
    <panel>

      <table>
        <title>Top Sourcetypes (Last 24 hours)</title>

        <!-- A search powers the panel -->
        <searchString>
          index=_internal | top limit=100 sourcetype | eval percent = round(percent,2)
        </searchString>

        <!-- Specify a time range for the search -->
        <earliestTime>-24h@h</earliestTime>
        <latestTime>now</latestTime>

        <!-- Use options to further define how to display result data -->
        <option name="wrap">true</option>
        <option name="rowNumbers">true</option>
      </table>
    </panel>
  </row>

</dashboard>
```

パネルの基盤となるサーチ

このダッシュボードは、以下のサーチを表しています。

- インラインサーチ
- レポートとして保存されたサーチ
- プレビルト・パネルからのサーチ
- ピボットから派生したインライン・サーチ

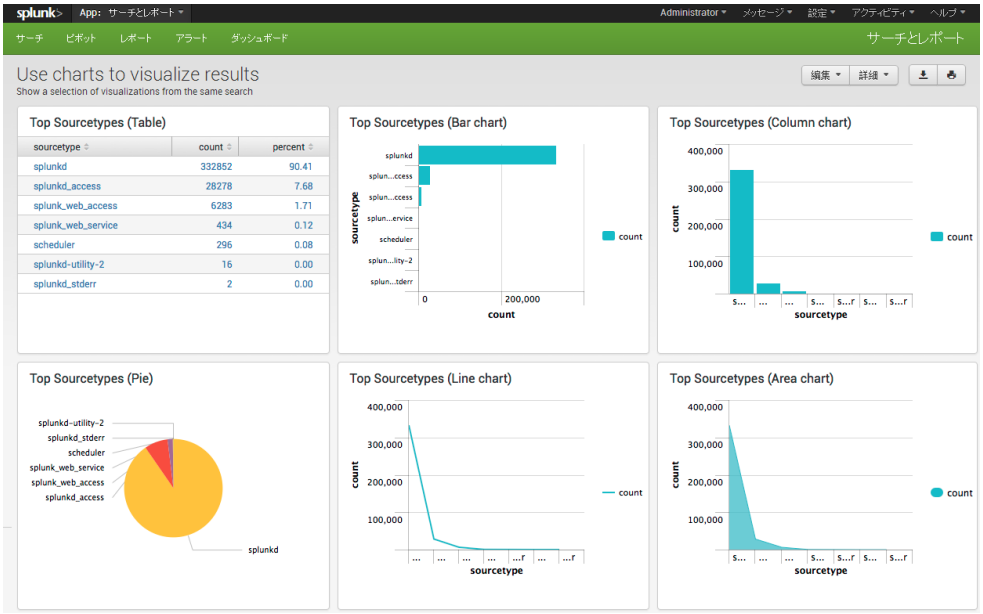



```
<dashboard>
  <label>Searches power dashboards</label>
  <description>Show the various searches to power a panel.</description>
  <!-- This row contains three panels -->
  <row>
```

```
    <panel>
      <table>
        <title>(Inline Search) Top Source Types</title>
        <!-- Inline Search -->
        <search>
          <query>
          index=_internal | top limit=100 sourcetype
          | eval percent = round(percent,2)
          </query>
          <earliest>-24h@h</earliest>
          <latest>now</latest>
        </search>
        <option name="rowNumbers">true</option>
      </table>
    </panel>
    <panel>
      <chart>
        <title>(Report) Top Source Types</title>
        <!-- Reference to a search saved as a report -->
        <search ref="Top Source Types Report" />
      </chart>
    </panel>
  </row>
  <row>
    <panel ref="top_source_types_in_the_last_hour" app="search" />

    <panel>
      <chart>
        <title>(Pivot)  Game Purchases</title>

        <!-- Inline search derived from a pivot -->
        <search>
          <query>
          | pivot Buttercup_Games Successful_purchases count(Successful_purchases)
          AS "Count of Successful purchases" SPLITROW product_name
          AS "product name" SORT 100 product_name
          </query>
        </search>
        <option name="charting.chart">pie</option>
      </chart>
    </panel>
  </row>
</dashboard>
```

パネルを使ったサーチ結果の視覚化

Splunk には、サーチ結果を表示するために利用できる各種視覚エフェクトが用意されています。結果はテーブルまたはイベントリストに表示できますが、さまざまなグラフを使用することもできます。グラフの種類を指定する <chart> エレメントと、子エレメントの <option> を使用します。




```
<dashboard>
  <label>Use charts to visualize results</label>
  <description>Show a selection of visualizations from the same search</description>
  <row>
    <panel>
      <!-- Display results as a table. Uses an           -->
      <!-- inline search, equivalent to the <searchName> -->
      <!-- specified for the other panels                -->
      <table>
        <title>Top Source Types (Table)</title>
        <search>
          <query>
            index=_internal | top limit=10 sourcetype
          </query>
          <earliest>-24h</earliest>
        <latest>now</latest>
        </search>
      </table>
    </panel>
    <panel>
      <!-- display same search as various charts -->
      <chart>
        <title>Top Source Types (Bar)</title>
        <search>
          <query>
            index=_internal | top limit=10 sourcetype
          </query>
          <earliest>-24h</earliest>
        <latest>now</latest>
        </search>
        <!-- specify the chart type with this <option> to <chart> -->
        <option name="charting.chart">bar</option>
        <option name="charting.axisY.scale">log</option>
      </chart>
    </panel>
    <panel>
      <chart>
        <title>Top Source Types (Column)</title>
        <search>
          <query>
            index=_internal | top limit=10 sourcetype
          </query>
          <earliest>-24h</earliest>
        <latest>now</latest>
        </search>
        <option name="charting.chart">column</option>
```

```
        <option name="charting.axisY.scale">log</option>
      </chart>
    </panel>
  </row>
  <row>
    <panel>
      <chart>
        <title>Top Source Types (Pie)</title>
        <search>
          <query>
            index=_internal | top limit=10 sourcetype
          </query>
          <earliest>-24h</earliest>
        <latest>now</latest>
        </search>
        <option name="charting.chart">pie</option>
      </chart>
    </panel>
    <panel>
      <chart>
        <title>Top Source Types (Line)</title>
        <search>
          <query>
            index=_internal | top limit=10 sourcetype
          </query>
          <earliest>-24h</earliest>
        <latest>now</latest>
        </search>
        <option name="charting.chart">line</option>
        <option name="charting.axisY.scale">log</option>
      </chart>
    </panel>
    <panel>
      <chart>
        <title>Top Source Types (Area)</title>
        <search>
          <query>
            index=_internal | top limit=10 sourcetype
          </query>
          <earliest>-24h</earliest>
        <latest>now</latest>
        </search>
        <option name="charting.chart">area</option>
        <option name="charting.axisY.scale">log</option>
      </chart>
    </panel>
  </row>
</dashboard>
```

リアルタイムサーチを使用するダッシュボード

Splunk のダッシュボードエディタを使って、またはシンプル XML でダッシュボードを記述して、リアルタイムダッシュボードを作成することができます。この例は、シンプル XML での記述方法を表しています。

リアルタイム・サーチを有効にするには、<search> エレメントで **<earliest>** および **<latest>** 子エレメントを使用します。たとえば、リアルタイムサーチを有効にして、データをテーブルに表示する場合は、以下のように指定します。

<table>

```
  <title>Look here for errors</title>
   <search>
     <query>
     error OR failed OR severe
     OR ( sourcetype=access_* ( 404 OR 500 OR 503 ) )
     </query>
   <earliest>rt</earliest>
   <latest>rt</latest>
   </search>
   <fields>host, source, errorNumber</fields>
```

</table>

リアルタイムダッシュボードの期間 (ウィンドウ) を設定することもできます。たとえば、過去 5 分間のリアルタ

イムイベントのみを表示することができます。

<table>

```
  <title>Look here for errors during the last 5 minutes</title>
   <search>
     <query>
       error OR failed OR severe OR ( sourcetype=access_* ( 404 OR 500 OR 503 ) )
     </query>
     <earliest>rt-5m</earliest>
     <latest>rt</latest>
   </search>
   <fields>host, source, errorNumber</fields>
```

</table>

サーチウィンドウの設定については、『サーチマニュアル』の「サーチへのリアルタイム時間範囲ウィンドウの指定」を参照してください。

グラフ内のフィールドへのカスタム色の指定

フィールドの色表示をカスタマイズするには、<chart> エレメントに charting.fieldColors 属性を使用します。これにより、グラフ内のフィールドを表すために使用する色を指定できます。ダッシュボード内で他のグラフや色指定とは関係なく、グラフの表示時には毎回指定した色が使用されます。

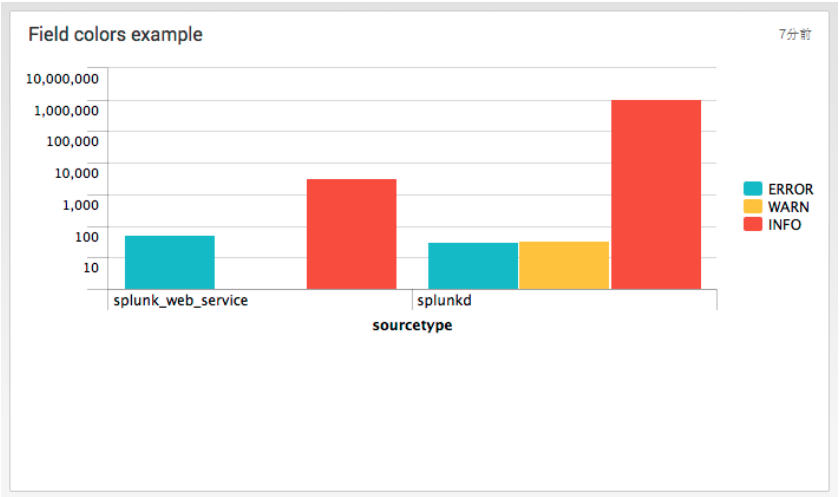
パネルエディタからフィールドの色を指定することはできません。charting.fieldColors 属性はシンプル XML コード内で使用します。『グラフ設定リファレンス』の「charting.fieldColors エントリ」を参照してください。

この例は、ソースタイプに対して描画されるエラーメッセージ数の色の指定方法を表しています。この例では、以下のサーチを使用します。

```
index = _internal log_level=* | stats count(eval(log_level="ERROR")) as ERROR count(eval(log_level="WARN")) as WARN
count(eval(log_level="INFO")) as INFO by sourcetype
```

fieldColors 属性は、結果内の特定のフィールドに適用されます。サーチは eval 式を使って、各ログレベルの結果を判断しています。次に、それらの結果に固有のフィールド名が作成されます。次に各ログレベルに対して、fieldColors 属性で色が割り当てられます。

以下の画像は、フィールドのデフォルト色を使ったベースサーチを表しています。

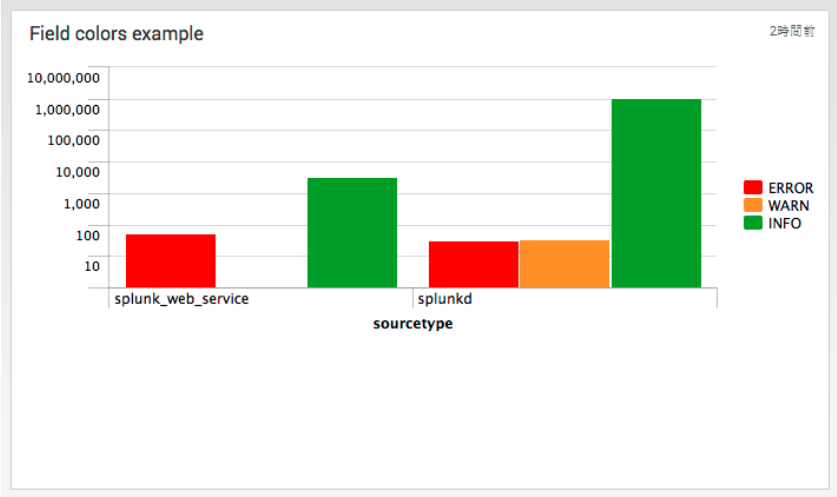


<chart> エレメントに以下のオプションを追加して、各ログレベルの色を定義します。

- INFO: green
- WARN: orange
- ERROR: red

```
<option name="charting.fieldColors">
  {"ERROR": 0xFF0000, "WARN": 0xFF9900, "INFO":0x009900, "NULL":0xC4C4C0}
</option>
```

以下の画像は、charting.fieldColors 属性を使って色を定義した、同じベースサーチを表しています。



グラフにカスタムのフィールド色を実装するコードを以下に示します。

```
<panel>
  <html>
    Use <tt>eval</tt> function in the search to transpose
    the value of the log_level field into individual fields
    for <tt>charting.fieldcolors</tt>.
  </html>
  <chart>
    <title>Field colors example</title>
    <search>
      <query>
        index = _internal log_level=* | stats
        count(eval(log_level="ERROR")) as ERROR
        count(eval(log_level="WARN")) as WARN
        count(eval(log_level="INFO")) as INFO
        by sourcetype
      </query>
      <earliest>-7d@h</earliest>
      <latest>now</latest>
    </search>
    <option name="charting.axisY.scale">log</option>
    <option name="charting.chart">column</option>
    <option name="charting.fieldColors">
      {"ERROR": 0xFF0000, "WARN": 0xFF9900, "INFO":0x009900, "NULL":0xC4C4C0}
    </option>
    <option name="charting.legend.placement">right</option>
  </chart>
</panel>
```

視覚エフェクトのプロパティの指定

シンプル XML は、すべての視覚エフェクトに適用できるプロパティを定義する一連のシンプル XML エレメントを提供しています。特定のタイプの視覚エフェクト (<char> や <map> など) に固有のプロパティについては、<option> エレメントを使ってプロパティを指定します。

特定のエレメントまたは <option> エレメントの使用方法はさまざまです。パネルのプロパティ指定の詳細は、「シンプル XML リファレンス」および「グラフ設定リファレンス」を参照してください。

すべての視覚エフェクトで利用できる、一部のエレメントの概要を以下の表に示します。

タグ	説明
<title>	文字列 パネルに「**Failed logins**」 (失敗したログイン) などのタイトルを追加します。タイトルはパネルの上部に表示されます。
<earliest> **<latest>**	Splunk 時間フォーマット サーチ結果を特定の時間ウィンドウに制限するには、earliest で開始して、latest で終了します。リアルタイムサーチを有効にする場合は、「rt」を指定します。

以下の <chart> エレメントを使ったパネルの例は、タイトルとインラインサーチの指定方法を表しています。ここでサーチ結果は 5 時間のウィンドウおよび 3 つのフィールドに制限されています。

```
<dashboard>
 <label>My dashboard</label>
  <row>
    <panel>

      <table>
        <title>Top users, five hours ago</title>
        <search>
          <query>
            host=production | top users
          </query>
        <earliest>-10h</earliest>
        <latest>-5h</latest>
        </search>
        <fields>host,ip,username</fields>
       </table>

    </panel>
  </row>
</dashboard>
```

以下の例は、<table> の <option> エレメントを使った、各種プロパティの指定方法を表しています。

```
<dashboard>
  <label>My dashboard</label>
  <row>
    <panel>

      <table>
        <title>Errors in the last 24 hours</title>
        <search>
          <query>
            Errors in the last 24 hours
          </query>
        </search>
        <option name="count">15</option>
        <option name="displayRowNumbers">true</option>
        <option name="maxLines">10</option>
        <option name="segmentation">outer</option>
        <option name="softWrap">true</option>
      </table>

    </panel>
  </row>
</dashboard>
```

以下の例は、X 軸と Y 軸の表示名を持つ縦棒グラフを示します。

```
<dashboard>
  <label>My dashboard</label>
  <row>
    <panel>
      <chart>
        <search>
          <query>
            sourcetype=access_* method=GET | timechart count by categoryId
            | fields _time BOUQUETS FLOWERS
           </query>
           <earliest>-7d</earliest>
           <latest>now</latest>
        </search>
        <title>Views by product category, past week (Stacked)</title>
        <option name="charting.axisTitleX.text">Views</option>
```

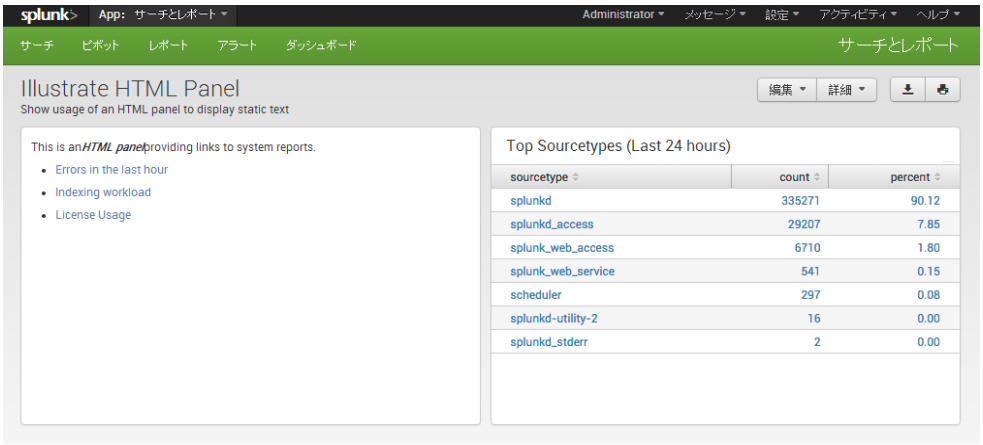
```
        <option name="charting.axisTitleY.text">Date</option>
        <option name="charting.chart">column</option>
      </chart>
    </panel>
  </row>
</dashboard>
```

HTML パネルを使った静的テキストの表示

HTML パネルには、インライン HTML が表示されます。ダッシュボードにドキュメント、リンク、画像、および他の Web コンポーネントを追加するには、HTML パネルを使用します。

Splunk は、HTML タグ間のコンテンツを、指定された HTML 書式設定に従って表示します。相対リンク参照は、現在のビューの場所からの相対的な位置となります。HTML パネルは、他の全般的なパネルオプションを使用せず、HTML 用に設定する特別なオプションもありません。

HTML パネルの使用方法の詳細は、「シンプル XML リファレンス」の「<html> エレメントエントリ」を参照してください。



この例で、アンカータグは次の特別な Splunk ロケーターを使ってシステムレポートにアクセスしています： `@go?s=`

```
. . .
<row>
  <panel>
    <html>
      <p>This is an <i><b>HTML panel</b></i> providing links to system reports.</p>
      <ul>
        <li>
          <p><a href="@go?s=Errors in the last hour">Errors in the last hour</a></p>
        </li>
        <li>
          <p><a href="@go?s=Indexing workload">Indexing workload</a></p>
        </li>
        <li>
          <p><a href="@go?s=License Usage Data Cube">License Usage</a></p>
        </li>
      </ul>
    </html>
  </panel>
    . . .
</row>
```

動的ドリルダウン機能を持つダッシュボードの設定

動的ドリルダウンにより、サーチ結果内のフィールドから他の Splunk ビューまたは Web ページへのリンクを指定することができます。ダッシュボードに動的ドリルダウンを実装するには、以下の手順に従ってください。

- サーチ結果を表示する視覚エフェクトに、`<drilldown>` タグを追加します。
- `<drilldown>` タグ内に、1 つまたは複数の `<link>` タグを追加します。
- 各 `<link>` タグ内に、リンク先の Splunk ビューまたは Web サイトを追加します。
- ドリルダウン操作に使用する結果の値を指定します。例：

- Splunk ビューのソースタイプとして使用するフィールド名を指定します。
- Web サイトに渡すことができる値を指定します。

詳細な例については、「ダッシュボードとフォームの動的なドリルダウン」を参照してください。

フォームの例

フォームはダッシュボードと類似の Splunk **ビュー**ですが、テキストボックス、ドロップダウンメニュー、ラジオボタンなどを使用して、値を 1 つまたは複数のサーチ単語に提供するためのインターフェイスをユーザーに提供しています。フォームを利用することで、ユーザーに複雑なサーチ言語を意識させることなく、目的の単語のみを使用して結果を得ることができます。結果は、テーブル、イベントリスト、またはダッシュボードで利用できる任意の視覚エフェクトに表示できます。

このトピックには、Splunk Enterprise のフォームの作成方法の基本的な例が記載されています。より複雑なソース・データを使用するその他の例については、Splunk 6.x Dashboard Examples App を参照してください。この例は、フォームでトークンを使って値を渡す例を表しています。トークンの導入の詳細は、「ダッシュボードでのトークンの使用」を参照してください。

基本的なフォームの例

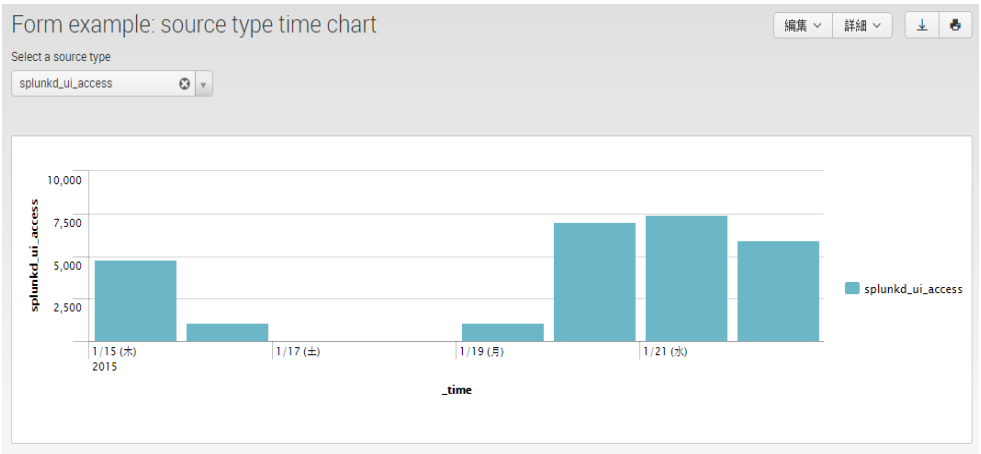
フォームへのユーザー入力は、入力の選択された値のトークンを定義しています。フォーム内のサーチはトークンを使って、サーチで使用する値を指定しています。サーチは、「`$...$`」をトークン値の区切り文字として使い、トークンの値にアクセスします。

たとえば、以下のコード・スニペットは、ユーザーの選択項目を表すために `sourcetype_tok` トークンを使用するドロップダウンを定義しています。また、ドロップダウンの選択項目も定義しています。

```
<input type="dropdown" token="sourcetype_tok">
   <label>Select a source type</label>
   <default>splunkd</default>
   <choice value="splunkd">splunkd</choice>
   <choice value="splunk_web_access">splunk_web_access</choice>
   <choice value="splunkd_ui_access">splunkd_ui_access</choice>
 </input>
```

フォーム内のサーチは、トークンを参照します。以下のコード・スニペットで、`$sourcetype_tok$` はドロップダウンの選択項目からの値を表しています。

```
    <search>
       <query>
         index = _internal sourcetype=$sourcetype_tok$
         | timechart count by sourcetype
       </query>
         <earliest>-7d</earliest>
         <latest>-0d</latest>
     </search>
```



シンプル XML を実装したフォームを以下に示します。

```
<form>
  <label>Form example: source type time chart</label>

  <!--autoRun means the search runs as soon as it is loaded. -->
  <!-- Do not need a submit button                          -->
```

```
  <fieldset autoRun="true" submitButton="false">
    <input type="dropdown" token="sourcetype_tok">
      <label>Select a source type</label>
      <default>splunkd</default>
      <choice value="splunkd">splunkd</choice>
      <choice value="splunk_web_access">splunk_web_access</choice>
      <choice value="splunkd_ui_access">splunkd_ui_access</choice>
    </input>
  </fieldset>

  <row>
    <panel>
      <chart>
        <search>
          <query>
            index = _internal sourcetype=$sourcetype_tok$
            | timechart count by sourcetype
          </query>
            <earliest>-7d</earliest>
            <latest>-0d</latest>
        </search>
      </chart>
    </panel>
  </row>
</form>
```

時間入力を持つフォームの例

フォームに 1 つまたは複数の時間入力を追加することができます。単一の時間入力を追加する場合、時間入力用のトークンは必要ありません。時間入力は、フォーム内のすべてのサーチにデータを提供します。

ただし、フォームに他の時間入力を追加した場合は、各時間入力にトークンを指定します。フォーム内のサーチは、トークンを参照して使用する時間入力を示します。

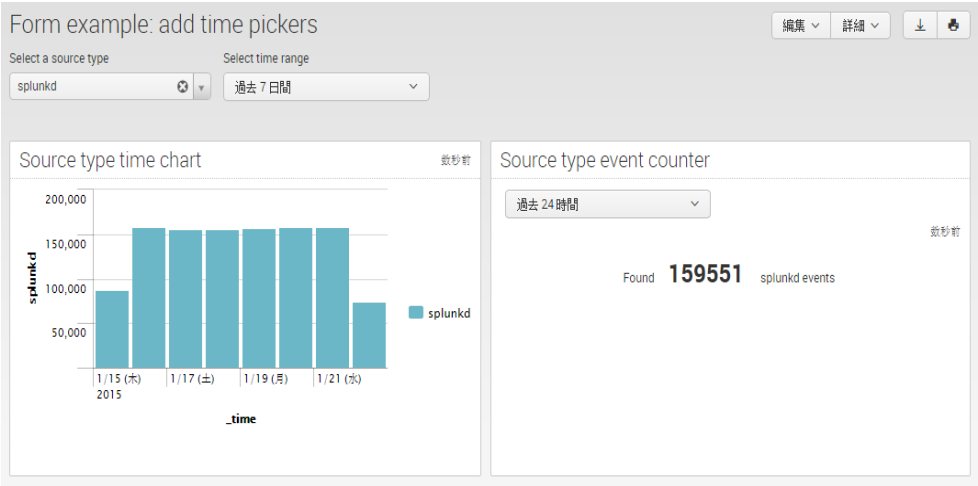
以下のコード・スニペットは、ローカルで使用するトークンを定義した時間入力を作成します。

```
    <input type="time" token="time_tok" searchWhenChanged="true">
       <label></label>
       <default>
         <earliest>-24h@h</earliest>
         <latest>now</latest>
       </default>
     </input>
```

ローカル時間入力にアクセスする場合、時間入力トークンに `earliest` および `latest` 修飾子を使用します。

```
      <search>
         <query>
           index=_internal sourcetype=$sourcetype_tok$
           | stats count as sourcetype</query>
         <earliest>$time_tok.earliest$</earliest>
         <latest>$time_tok.latest$</latest>
       </search>
```

[Source Type Timechart] パネル用のグローバル・タイマーを使用する例を以下に示します。[Source Type Timechart] パネルには、そのパネルのローカル時刻のみが含まれています。




```
<form>
  <label>Form example: add time pickers</label>
  <fieldset autorun="true" submitButton="false">
    <input type="dropdown" token="sourcetype_tok">
      <label>Select a source type</label>
      <default>splunkd</default>
      <choice value="splunkd">splunkd</choice>
      <choice value="splunk_web_access">splunk_web_access</choice>
      <choice value="splunkd_ui_access">splunkd_ui_access</choice>
    </input>

    <!-- Global timer. Not token is necessary -->
    <input type="time" searchWhenChanged="true">
      <label>Select time range</label>
      <default>
        <earliest>-7d@h</earliest>
        <latest>now</latest>
      </default>
    </input>

  </fieldset>
  <row>
    <panel>
      <title>Source type time chart</title>
      <chart>
        <search>
          <query>index = _internal sourcetype=$sourcetype_tok$
            | timechart count by sourcetype</query>
        </search>
      </chart>
    </panel>
    <panel>
      <title>Source type event counter</title>

      <!-- Local timer. Use tokens to access selected time. -->
      <input type="time" token="time_tok" searchWhenChanged="true">
        <label></label>
        <default>
          <earliest>-24h@h</earliest>
          <latest>now</latest>
        </default>
      </input>

      <single>
        <search>
          <query>
            index=_internal sourcetype=$sourcetype_tok$
            | stats count as sourcetype</query>

          <!-- Use the earliest and latest modifiers to the time input token -->
          <earliest>$time_tok.earliest$</earliest>
```

```
        <latest>$time_tok.latest$</latest>

      </search>
      <option name="beforeLabel">Found </option>
      <option name="afterLabel">$sourcetype_tok$ events</option>
    </single>
  </panel>
 </row>
</form>
```

フォームへの静的および動的入力

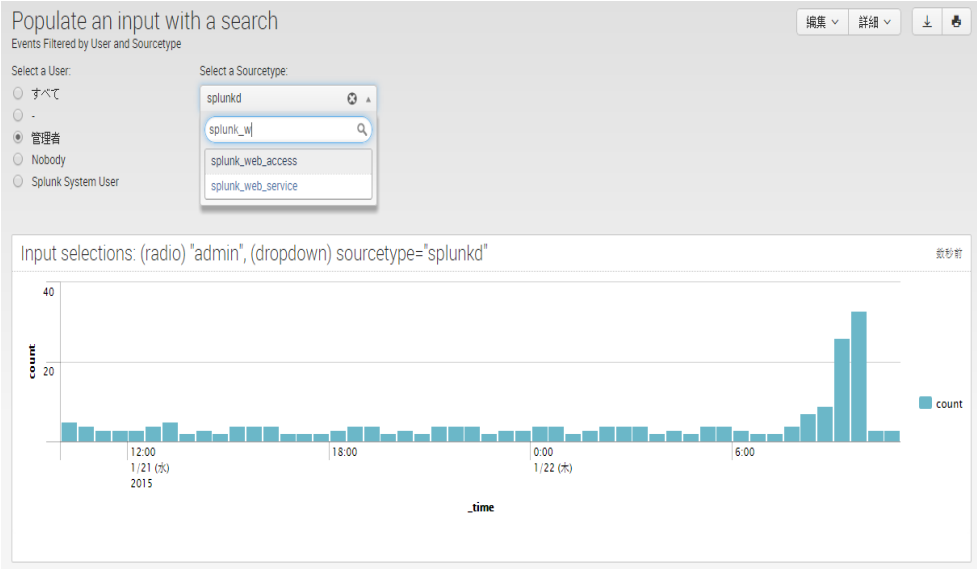
以下のフォーム入力には、ユーザー選択用の複数の選択項目が必要です。フォームには、静的に入力を定義、またはサーチを使って動的に入力を設定することができます。

- チェックボックス
- ドロップダウン
- 複数選択
- ラジオボタン

以下の例のサーチは、選択項目の静的定義と動的定義を比較しています。ドロップダウンは、設定用サーチを使って選択項目を定義しています。

- 設定用 <search>
  選択項目のラベルと値に使用するフィールドを返します。

- <fieldForLabel> <fieldForValue>
  <input> エレメントの子エレメント。これらは、ドロップダウンの選択項目を設定するために使用するフィールドを指定しています。



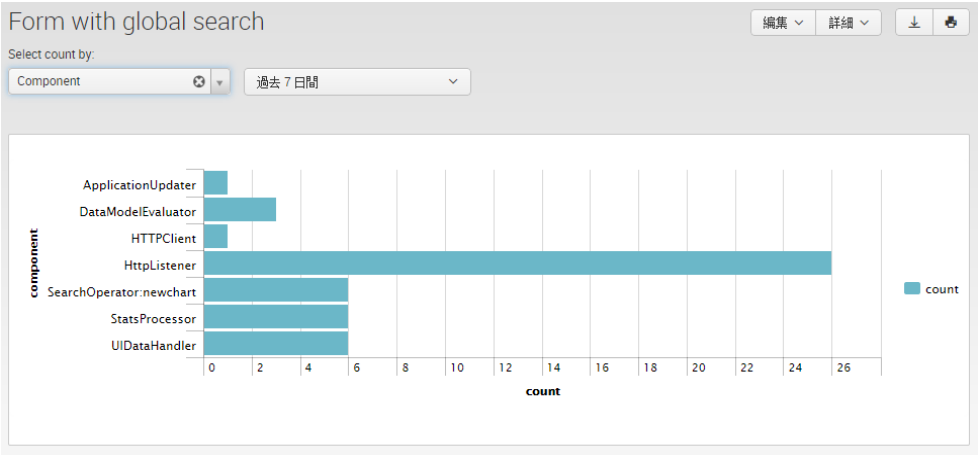

```
<form>
  <label>Populate an input with a search</label>
  <description>Events Filtered by User and Sourcetype</description>
  <!-- Do not need a Search Button. Inputs search when changed -->

  <fieldset autoRun="true" submitButton="false">

    <!-- Static definition of choices -->
    <input type="radio" token="username_tok" searchWhenChanged="true">
      <label>Select a User:</label>

      <!-- Define the default value -->
      <default>All</default>

      <!-- Hard-code the choices -->
      <choice value="*">All</choice>
      <choice value="-">-</choice>
      <choice value="admin">Admin</choice>
      <choice value="nobody">Nobody</choice>
      <choice value="splunk-system-user">Splunk System User</choice>
    </input>
```

```
    <!-- Dynamic definition of choices -->
    <input type="dropdown" token="sourcetype_tok" searchWhenChanged="true">
      <label>Select a Sourcetype:</label>
      <prefix>sourcetype="</prefix>
      <suffix>"</suffix>

      <!-- Define the default value -->
      <default>splunkd</default>

      <!-- Hard-code the choice for "All" -->
      <choice value="*">All</choice>

      <!-- Define the other choices with a populating search -->
      <search>
        <query>
          index=_internal | stats count by sourcetype
        </query>
      </search>
      <fieldForLabel>sourcetype</fieldForLabel>
      <fieldForValue>sourcetype</fieldForValue>
    </input>

  </fieldset>
  <row>
    <panel>
      <!-- Use tokens from the <input> elements in the panel title -->
      <title>
        Input selections: (radio) "$username_tok$", (dropdown) $sourcetype_tok$
      </title>

      <chart>

        <!-- search for the visualization, references the input tokens-->
        <search>
          <query>
            index=_internal user=$username_tok$ $sourcetype_tok$ | timechart count
          </query>
          <earliest>-24h@h</earliest>
          <latest>now</latest>
        </search>
       </chart>

    </panel>
  </row>
</form>
```

グローバルサーチを使ったフォームの作成

さまざまなパネルにデータを提供する、グローバル・サーチを使用したフォームを作成することができます。このシナリオは、別の形式の後処理サーチです。後処理サーチにはさまざまな制限があるため、使用する場合は注意する必要があります。多くの場合、後処理サーチが常にサーチ・リソースの使用効率を向上するとは限りません。「後処理サーチ」を入念に参照してください。ここでは、後処理の制限事項および後処理サーチを利用する前に検討しておく必要がある他の要因を説明しています。

グローバル・サーチを利用したフォームの例を以下に示します。



このグローバル・サーチは変換サーチ・コマンドを使って、後処理サーチに渡すイベント数に関する 10,000 件の制限を回避しています。

```
<search id="global_search">
   <query>
     index=_internal source=*splunkd.log | stats count by component, log_level
   </query>
 </search>
```

ドロップダウン選択項目の値には、後処理サーチが含まれています。

```
<fieldset autoRun="true" submitButton="false" searchWhenChanged="true">
   <input type="dropdown" token="stats_tok">
     <label>Select count by:</label>
     <default>Log level</default>
     <choice value="stats sum(count) AS count by log_level">Log level</choice>
     <choice value="search log_level=error | stats sum(count) AS count by component">Component</choice>
   </input>
```

フォーム内のパネルは、ドロップダウンのトークンを使って選択された項目にアクセスします。

```
      <search base="global_search">
         <query>
           $stats_tok$
         </query>
       </search>
```

グローバル・サーチを使ったフォームの、全コードを以下に示します。

<form>

```
<label>Form with global search</label>
 <search id="global_search">
   <query>
     index=_internal source=*splunkd.log | stats count by component, log_level
   </query>
 </search>

 <fieldset autoRun="true" submitButton="false" searchWhenChanged="true">
   <input type="dropdown" token="stats_tok">
     <label>Select count by:</label>
     <default>Log level</default>
     <choice value="stats sum(count) AS count by log_level">Log level</choice>
     <choice value="search log_level=error | stats sum(count) AS count by component">Component</choice>
   </input>

   <input type="time">
     <default>Last 7 days</default>
   </input>
 </fieldset>
<row>
   <panel>
     <chart>
       <option name="charting.chart">bar</option>
       <search base="global_search">
```

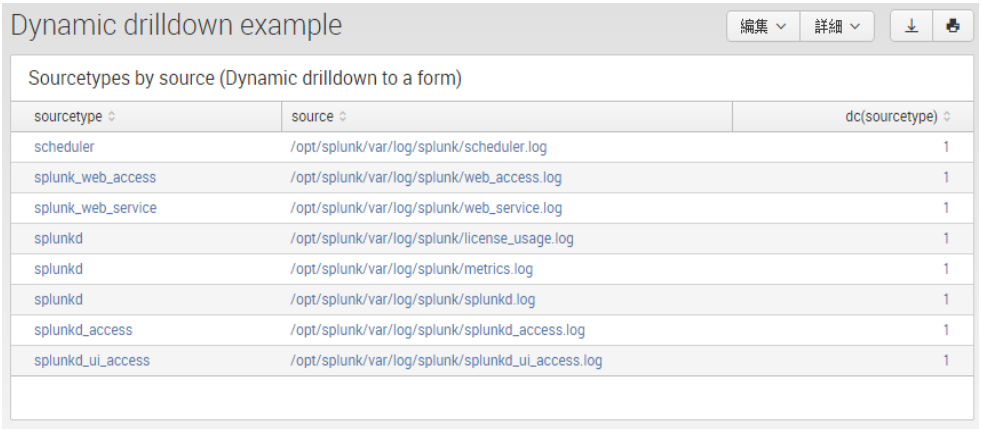
```
        <query>
          $stats_tok$
        </query>
      </search>
    </chart>
  </panel>
 </row>
</form>
```

ダッシュボードとフォームの動的なドリルダウン

動的なドリルダウンを利用して、ユーザーがダッシュボードやフォーム内のフィールドをクリックした時のカスタム宛先リンクを定義できます。クリックで捕捉された値は、宛先に渡されます。ご利用の Splunk Enterprise インストール内の他のダッシュボード、フォーム、またはビューを宛先として利用できます。外部の Web ページを宛先にすることもできます。

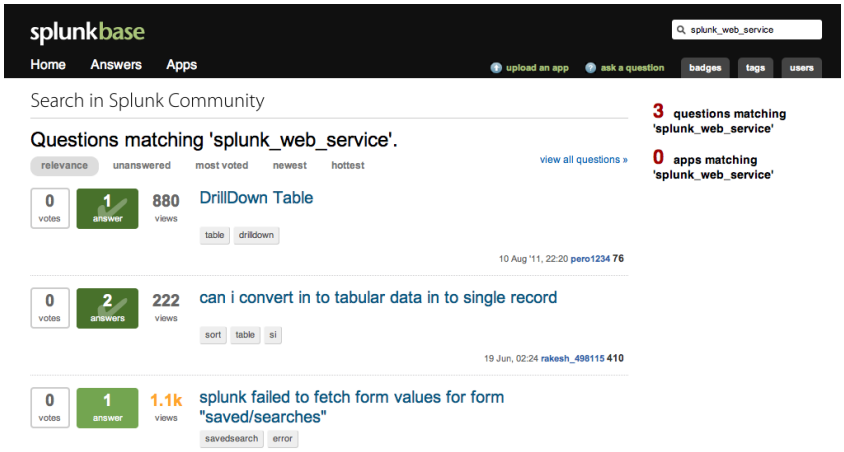
> **注意：**Splunk Enterprise には、即座に利用できる基本的なドリルダウン機能が用意されています。ドリルダウンの主要機能の仕組みについては、このマニュアルの「ドリルダウン動作」を参照してください。「ドリルダウン動作」には、概念的な情報および動的ドリルダウンの例も含まれています。

たとえば、以下のダッシュボードにはソースタイプのスループットがテーブルとして表示されます。以下の図は、選択したセル「splunk_web_service」をクリックした、動的ドリルダウンの結果を表しています。



他のフォームを開く宛先を定義して、クリックされたソースタイプをそれに渡すことができます。これは上記のダッシュボードで **splunk_web_service** をクリックした結果です。

また、クリックされた値を Splunk Answers などの Web ページに渡すこともできます。



動的ドリルダウンの基本

動的ドリルダウンを実装するには、<drilldown> タグを使用します。
テーブルまたはグラフ内に <drilldown> タグを配置します。

<drilldown> タグ内に、必要に応じて `target="[attribute]"` を指定して、ドリルダウン宛先を指示します。この属性のデフォルトは `target="_self"` で、この場合現在のウィンドウにリンクが開かれます。

<drilldown> タグの間に、1 つまたは複数の <link> タグを追加します。<link> タグを使ってドリルダウンの宛先を指定します。例：

```
<dashboard>
  <label>Dynamic drilldown example</label>

  <row>
    <panel>
      <table>

      <title>Sourcetypes by source (Dynamic drilldown to a form)</title>
      <search>
        <query>
        index="_internal" | stats dc(sourcetype) by sourcetype, source
        </query>
      <earliest>-60m</earliest>
      <latest>now</latest>
      </search>
      <option name="count">15</option>
      <option name="displayRowNumbers">false</option>
      <option name="showPager">true</option>

      <drilldown target="My New Window">
       <!-- Access the input on the target form, which is in the same app  -->
       <!-- sourcetype.tok is the token for an input to the target form    -->
       <link>
          form_for_drilldown?form.sourcetype_tok=$click.value$
        </link>
      </drilldown>

      </table>
    </panel>
  </row>

</dashboard>
```

宛先の指定

リンクを指定するための構文を以下に示します。

<drilldown>

```
 <link>...</link>
  <link>...</link>
```

```
   . . .
  <link>...</link>

</drilldown>
```

<link> タグで宛先を指定するには、さまざまな方法があります。ここでは、さまざまな状況での宛先を指定する構文の例を説明していきます。

1.相対パスを使ってダッシュボードに接続します。
2.相対パスを使ってフォームに接続し、トークンを渡してフォームに記入します。
3.元のサーチから、もっとも早い時間ともっとも遅い時間範囲を渡します。
　(次のセクションで説明するように CDATA を使用する必要があります。)
4.URL とクエリー引数を使って、宛先ページに値を渡します。

1) <link> *path*/*viewname* </link>
2) <link> *path*/*viewname*?form.*token*=$*dest_value*$ </link>
3) <link> *path*/*viewname*?form.*token*=$*dest_value*$&earliest=$earliest$&latest=$latest$ </link>
4) <link> *URL*?q=$*dest_value*$ </link>

値の捕捉

ダッシュボードやフォームから値を捕捉して、それを宛先に渡すためのさまざまな方法があります。

捕捉する値を指定するには、<condition> エレメントに `field` または `series` 属性を使用します。テーブルの場合は、`field` 属性を指定して、指定した列または行の値を捕捉します。グラフの場合は、`series` 属性を指定して、指定したシリーズの値を捕捉します。

たとえば、ダッシュボードに列 `A`、`B`、`C` があるテーブルがある場合に、以下の例を考えてみましょう。

1.列 A でクリックされた値を捕捉し、その値を使うフォームを開きます。列 A または 列 B のクリックには、デフォルトのドリルダウン動作を使用します。

<drilldown>

```
<condition field="A">
   <link> path/viewname?form.token=$dest_value$ </link>
 </condition>
```

</drilldown>

2.上記の 1 と同じ動作ですが、列 B をクリックすると、値を Web ページにクエリー引数として渡します。

<drilldown>

```
<condition field="B">
   <link>URL?q=$dest_value$</link>
 </condition>
```

</drilldown>

宛先を指定するための構文

宛先の値を指定する構文は、使用するグラフのタイプと選択した宛先によって異なります。詳細は、「シンプル XML リファレンス」の <drilldown> エレメント と <link> エレメントを参照してください。

動的ドリルダウンの例

ここでは、ダッシュボードやフォームでの、動的ドリルダウンの作成方法について説明していきます。サーチの大半は、『サーチチュートリアル』から利用できるデータにアクセスしています。これらの例のダッシュボードを作成するために『サーチチュートリアル』からデータをダウンロードする場合は、「Splunk Enterprise へのチュートリアルデータの取り込み」を参照してください。

動的ドリルダウンは、事前定義されたドリルダウン・トークンによって異なります。「ドリルダウンのトークンの定義」を参照してください。

宛先フォーム

これらの例は、デフォルトの Splunk サーチ App に関連する以下のフォームを作成したことを前提にしています。このフォームが、例の宛先フォームになります。

ドリルダウンの宛先フォーム：/app/search/form_for_drilldown

```
<form>
  <label>Destination form for drilldown</label>
  <fieldset autorun="true" submitButton="false">
    <input type="dropdown" token="sourcetype_tok">
      <label>Select a source type</label>
      <default>splunkd</default>
```

```
      <search>
        <query>
          index = _internal | stats count by sourcetype
        </query>
      </search>
      <fieldForLabel>sourcetype</fieldForLabel>
      <fieldForValue>sourcetype</fieldForValue>
    </input>
  </fieldset>
  <row>
    <panel>
      <chart>
        <search>
          <query>index = _internal sourcetype=$sourcetype_tok$
            | timechart count by sourcetype</query>
          <earliest>-7d</earliest>
          <latest>-0d</latest>
        </search>
      </chart>
    </panel>
  </row>
</form>
```

フォームにリンクするダッシュボード

この例は、テーブルから Splunk Enterprise フォームへのドリルダウンを実装した、ダッシュボードの使用方法を表しています。

この例が正常に動作する鍵となるのが、<link> タグです。このタグは以下の事項を指定します。

- ターゲットフォーム FormSearchDrillDown へのパス
- ターゲットで使用するトークン、sourcetype
- 選択された行のプロセッサ (processor) フィールドの値を、宛先フォームに渡します。このダッシュボードでは、行のどの部分をクリックしても、その行のプロセッサに対する値が取得されます。
- ターゲットビューに、サーチのもっとも早い時間ともっとも遅い時間を渡します。

注意：「&」文字が正しく解釈されるようにするために、CDATA セクションを使用してください。

```
<link>
<![CDATA[
  /app/search/form_for_drilldown?form.sourcetype=$row.sourcetype$&earliest=$earliest$&latest=$latest$
]]>
</link>
```

完成したダッシュボードコードを以下に示します。

Splunk フォームにリンクするダッシュボード

```
<dashboard>
 <label>Dashboard with dynamic drilldown to a Splunk form</label>
  <row>

    <table>
      <search>
        <query>
           index="_internal" group="per_sourcetype_thruput" |
           chart sum(kbps) over series
        </query>
        <earliest>-60m</earliest>
        <latest>now</latest>
      </search>
      <title>Top sourcetypes (drilldown example)</title>
      <option name="count">15</option>
      <option name="displayRowNumbers">false</option>
      <option name="showPager">true</option>

     <drilldown>
       <link>
       <![CDATA[
  /app/search/form_for_drilldown?form.sourcetype=$row.sourcetype$&earliest=$earliest$&latest=$latest$
       ]]>
       </link>
```

```
    </drilldown>
   </table>

  </row>
</dashboard>
```

Splunk Answers Web サイトにリンクするフォーム

この例は、グラフから外部 Web サイトへのドリルダウンを実装したフォームの使用方法を表しています。

この例が正常に動作する鍵となるのが、<link> タグです。このタグは以下の事項を指定します。

- Splunk Answers へのフル URL
- $click.value$ を使って X 軸から値を取得し、それを Splunk Answers にクエリーパラメータとして渡します

```
<link>
  http://splunk-base.splunk.com/integrated_search/?q=$click.value$
</link>
```

フォームの完成したコードを以下に示します。

外部 Web サイトのリンクへの動的ドリルダウンを使用する Splunk フォーム

```
<form>
  <label>Form Search (Beta)</label>

  <fieldset>
     <!-- Use the html tag to specify text to display -->
     <html>
       <p>Enter a sourcetype in the field below. This view returns the most recent 1000 events for that
sourcetype.</p>
       <p>In the Matching Events, click in the series column to open the value clicked in a new form</p>
     </html>

     <!-- The default input is a text box, with no seed value -->
     <input token="sourcetype" />

     <!-- Include a time picker -->
     <input type="time">
        <default>Last 30 days</default>
      </input>
  </fieldset>

  <row>
    <panel>
      <!-- output the results as a 50 row events table -->
      <table>
         <title>Matching events</title>

        <!-- search with replacement token delimited with $ -->
        <search>
          <query>
            index="_internal" group="per_sourcetype_thruput" series=$sourcetype$
            | chart sum(kbps) over series
          </query>
        </search>

         <option name="count">50</option>

         <!-- $click.value$ captures the value clicked by the user -->
         <!-- and passes it to the website as a query parameter        -->
         <drilldown>
           <link>
              http://splunk-base.splunk.com/integrated_search/?q=$click.value$
           </link>
         </drilldown>
       </table>
    </panel>
  </row>
```

```
</form>
```

複数値フィールドへのダッシュボードリンク

複数値フィールドを表示するダッシュボードがある場合、クリックされた値に固有のドリルダウン場所を指定できます。複数値フィールドは、イベント内に複数回登場するフィールドで、それぞれが異なる値を保有しています。複数値フィールドの詳細は、「複数値フィールドの設定」を参照してください。

一般的にテーブルの値では、`$click.name$` または `$click.name2$` を指定して、列または行からのドリルダウン値を収集します。ただし、複数値フィールドの場合は、`$click.value2$` を使用して、ドリルダウンの選択値を収集します。また、<condition> エレメントは `field` 属性を使って、列の選択項目を複数値フィールドに制限します。

たとえば、ダッシュボードの badges 複数値フィールドの、クリックされた値を収集する方法を以下に示します。このダッシュボードで、badges は Splunk 2012 Users Conference 中の FourSquare イベントへのユーザーのチェックイン数を表しています。

<drilldown>

```
<condition field="badges">
    <link>
      /app/foursquare_vegas/vegas_badge_1?form.badge=$click.value2$
   </link>
 </condition>
```

</drilldown>

- `field`：
  このフィールドに選択項目を制限します。
- /app/foursquare_vegas/vegas_badge_1
  ドリルダウンアクションのターゲットフォーム
- `form.badge`：
  クリックされた値に対してターゲットフォーム内で使用するトークン

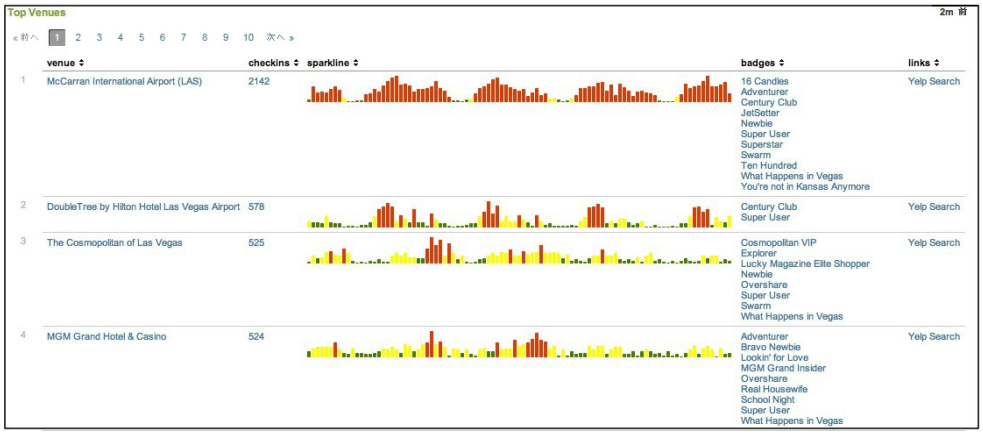
このダッシュボードの完全なソースコードを以下に示します。このダッシュボードには、他にも 2 つのドリルダウンリンクがあり、またスパークラインも実装されています (『サーチマニュアル』の「サーチ結果へのスパークラインの追加」を参照)。

複数値フィールドのドリルダウンは、コード内で呼び出されています。

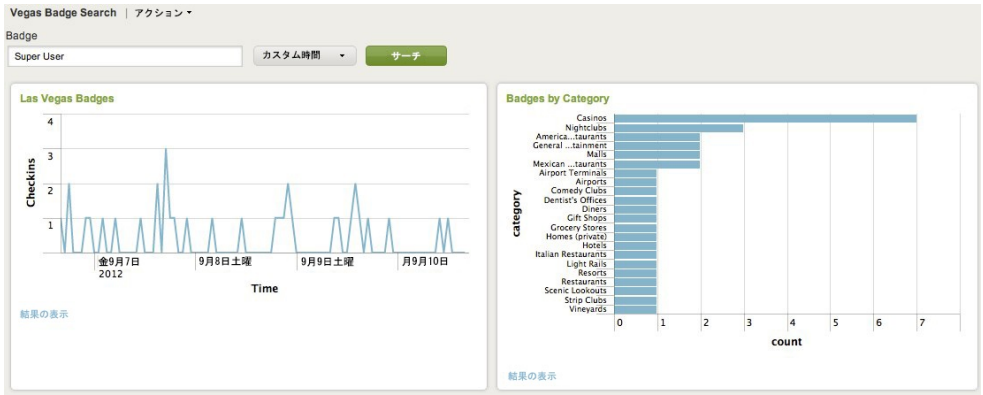
```
<!-- Dashboard enabling drilldown for a multivalue field -->

<dashboard>
  <label>Demo: drilldown</label>
  <row>
    <panel>
    <table>
      <searchString>
        index=foursquare checkin.primarycategory.nodename=*
        | spath output=venue path=checkin.venue.name
        | spath output=badges path=checkin.badges{}.name
        | eval link="Yelp Search"
        | stats count as checkins sparkline values(badges)
              as "badges" values(link) as "links" by venue
        | sort -checkins
      </searchString>

      <format field="sparkline" type="sparkline">
        <option name="type">bar</option>
        <option name="height">30</option>
        <option name="barColor">green</option>
        <option name="colorMap">
          <option name="5:9">yellow</option>
          <option name="10:">red</option>
        </option>
      </format>
      <title>Top Venues</title>

      <drilldown>

         <!-- Mulitvalue field drilldown -->
         <condition field="badges">
           <link >
             /app/foursquare_vegas/vegas_badge_1?form.badge=$click.value2$
```

```
          </link>
        </condition>

        <condition field="venue">
          <link>
            /app/foursquare_vegas/vegas_venue_1?form.venue=$row.venue$
          </link>
        </condition>

        <condition field="links">
         <link>
           http://www.yelp.com/search?find_desc=$row.venue$&find_loc=Las+Vegas,+NV
         </link>
        </condition>
      </drilldown>

    </table>
    </panel>
  </row>
</dashboard>
```

2012 Splunk Users Conference でデモンストレーションされた、実際のダッシュボードの例を以下に示します。このダッシュボードは Splunk 5 で表示されますが、その動的ドリル ダウンは Splunk 6 にも適用されます。



badges の値をクリックすると表示されるフォームを以下に示します。



ダッシュボードでのトークンの使用

シンプル XML ダッシュボードのサーチは、トークンを一種の変数として使用し、フィールド、フィールド値、およびサーチで使用する用語を動的に指定します。トークンは、フォームからの入力値の取得、動的ドリルダウン操作の導入、およびダッシュボード内のパネルの条件表示の指定に役立ちます。

トークンの概要

トークンは、ダッシュボード内で動的に値を渡すための手段です。これらの値はサーチが利用し、特別な構文を使ってトークンの値にアクセスします。トークンの値は、フォーム入力、視覚エフェクト用に事前定義されたトークン値など、さまざまなソースから取得されます。

トークンの値にアクセスするための基本的な構文は、$...$ 区切り文字を使用します。たとえば、以下の視覚エフェクト用サーチは、field_tok トークンにアクセスします。フォーム入力は、以前に field_tok トークンを定義しています。

```
index=_internal source=*splunkd.log | stats count by $field_tok$
```

トークン値にアクセスするための、高度な構文については、「トークン・フィルタ」を参照してください。

トークン値の生成

トークン値を生成するには、さまざまな方法があります。これには、以下の事項が含まれています。

- フォームの入力値を取得するトークンを定義する。
- トークンの値に基づいて、条件付きのアクションを指定するトークンを定義する。
- 前に定義されたトークンに基づく値を使用するトークンを、サーチ文字列内に定義する。
- Splunk Enterprise には、ユーザーが利用できるトークン値が定義されています。視覚エフェクト用トークン、時間入力用トークン、フォーム入力のラベルと値などのトークンが定義されています。

トークン値の使用

トークンの値にアクセスするための、さまざまな使用事例があります。

使用事例	説明
フォーム入力	フォームへの入力が、視覚エフェクトが表示するデータを変更します。ユーザー入力で定義されたトークンが、フォームのサーチを変更します。
フォーム内の複数のタイム・ピッカー	複数のタイム・ピッカーがあるフォームの場合、各視覚エフェクトで使用するタイム・ピッカーをトークンが示します。
動的ドリルダウン	ユーザーがダッシュボード内の視覚エフェクトをクリックすると、事前定義されたトークンが、ドリルダウンのためにクリックされた値を取得します。
ダッシュボード・エレメントの条件による表示	パネルのそのコンテンツの表示用の、トークン設定/設定解除条件。
グラフ内の領域を選択するための、パン/ズーム・グラフ・コントロール。	事前定義トークンにより、この動作の特定の領域を選択することができます。

SplunkJS スタックでのトークン

SplunkJS Stack と JavaScript エクステンションを使用する場合は、Splunk Developer Portal の「Tokens and Data Binding」を参照して、JavaScript でのトークンの使用方法を学習してください。

フォーム入力のトークンの定義

すべてのフォーム入力に、入力のユーザー選択値用トークンを定義するトークン属性があります。フォーム入力には、トークンの値をさらに変更する子エレメント <prefix> および <suffix> も用意されています。複数選択オプションの場合、トークンの値を変更できる追加のエレメントが存在しています。「複数選択入力のトークンの定義」を参照してください。

このコード・スニペットは、ドロップダウン・リスト用のトークンを定義しています。ドロップダウンで選択された項目が、トークンの値になります。

```
<input type="dropdown" token="sourcetype_tok">
  <label>Select a source type</label>
  <default>splunkd</default>
  <choice value="splunkd">splunkd</choice>
  <choice value="splunk_web_access">splunk_web_access</choice>
  <choice value="splunkd_ui_access">splunkd_ui_access</choice>
</input>
```

「フォーム入力の例」を参照してください。

複数選択入力のトークンの定義

複数選択入力は <prefix>、<suffix>、<valuePrefix>、<valueSuffix>、および <delimiter> エレメントを使って、選択された項目の複数選択サーチを作成します。入力用トークンの値である複数選択サーチにより、入力が選択されたすべての値を、フォームのサーチに渡すことが保証されます。

以下のコード・スニペットは、複数選択トークンの値の作成方法を表しています。複数選択入力で、ユーザーが **splunkd** と **splunk_web_access** の両方を選択した場合、トークン値は以下のサーチ・フラグメントになります。

```
(sourcetype ="splunkd") OR (sourcetype ="splunk_web_access")
```

このサーチ・フラグメントは、以下のものから派生しています。

```
<prefix> + <valuePrefix> + [choice value] + <valueSuffix> + <suffix> + <delimiter> . . .
    (      sourcetype ="      splunkd              "            )        _OR_

    <input type="multiselect" token="sourcetype_tok">
      <label>Select one or more source types</label>

      <choice value="splunk_web_access">splunk_web_access</choice>
      <choice value="splunkd">splunkd</choice>
      <choice value="splunk_ui_access">splunk_ui_access</choice>
      <choice value="splunkd_access">splunkd_access</choice>

      <!--      Build multi-selection search:
        (sourcetype ="value1" OR sourcetype ="value2" OR ...)
      -->
      <prefix>(</prefix>
      <valuePrefix>sourcetype ="</valuePrefix>
      <valueSuffix>"</valueSuffix>
      <delimiter> OR </delimiter>
      <suffix>)</suffix>

    </input>
```

「複数選択入力の例」を参照してください。

時間入力のトークンの定義

異なるタイム・ピッカーを使用するパネルを持つフォームがある場合、時間入力用のトークンを使って各パネルで使用するタイム・ピッカーを指定します。タイム・ピッカーでもっとも早い時間ともっとも遅い時間の値にアクセスするには、トークンで以下の修飾子を使用します。

- $timer_tok.earliest$
- $timer_tok.latest$

トークンを定義しない時間入力はグローバルです。そのようなタイム・ピッカーで選択された値は、タイム・ピッカーを指定しないすべての視覚エフェクトに適用されます。

「時間入力の例」を参照してください。

フォーム入力による条件付き操作用トークンの定義

フォーム入力の条件付き操作用トークンを定義することができます。トークンの値は、指定した条件によって変化します。たとえば、トークンの条件値に基づいてサーチを変更、または表示する別の視覚エフェクトを選択することができます。

条件による操作には、以下の事項が含まれています。

- トークン値に基づいてサーチを変更する。
- 条件に基づいて、パネルとそのコンテンツを非表示、または表示します。
- トークン値に基づいて、開くビューを選択します。

条件付き操作は、フォーム入力と動的ドリルダウンで利用できます。フォーム入力は、以下のエレメントをさまざまな組み合わせで使用します。

エレメント	説明
<change>	定義した条件のコンテナ・エレメント。
<condition>	入力選択値に基づく条件を設定します。「条件入力の例」で、これはドロップダウン・リストで選択された項目の値です。
<link>	条件に基づいて、宛先へのリンクを指定します。
<set>	トークン用の各種値を定義します。ダッシュボードの <search> エレメントは、このトークンの値を使用します。 「条件入力の例」では、トークン `earliest_tok` の値を定義しています。
<unset>	前に設定されたトークンを削除します。 これは、設定されているトークンに基づく条件付き操作で役立ちます。

「フォーム入力による条件付き操作」の例を参照してください。

フォーム入力のラベルと値にアクセスする事前定義トークン

Splunk Enterprise には、フォーム入力のラベルと値にアクセスする、事前定義トークンが用意されています。トークンは、以下の入力で利用することができます。

- チェックボックス
- ドロップダウン・リスト

- 複数選択
- ラジオ・ボタン

トークン	説明
label	フォーム入力選択項目の、指定された名前を持っています。
value	フォーム入力選択項目の、値を持っています。

これらのトークンは、サーチのカスタマイズや、パネル/視覚エフェクトのタイトルや説明の選択項目のラベルの設定に役立ちます。

「フォーム入力のラベルと値へのアクセス」を参照してください。

動的ドリルダウンのトークンの定義

動的ドリルダウンの事前定義トークン

Splunk Enterprise には、動的ドリルダウン用の事前定義トークンが用意されています。事前定義トークンは、視覚エフェクト内でユーザーがクリックした場所に応じた値を取得します。「ダッシュボードとフォームの動的なドリルダウン」を参照してください。

利用できる事前定義トークン、およびそれが取得する値は、視覚エフェクトのタイプによって異なります。テーブル視覚エフェクトで利用できる事前定義トークンの一覧を以下の表に示します。『シンプル XML リファレンス』の「ドリルダウン・イベント・トークン」には、動的ドリルダウン用のすべての事前定義トークンが記載されています。

トークン	説明
click.name	テーブルに表示されている一番左のフィールドの名前。存在する場合は、常に `_time` になります。
click.value	行の一番左側の列の値。
click.name2	列名。
click.value2	列の値。
row.<fieldname>	表示されていないフィールドも含めた、行のすべてのフィールド値。
earliest/latest	テーブル行の時間範囲、または存在しない場合は、サーチの時間範囲。

<link> エレメントは、事前定義トークンの値を使って、新しいビューまたは Web ページにリンクします。「フォーム入力のラベルと値にアクセスする事前定義トークン」を参照してください。事前定義トークンは、<drilldown> エレメントを使った条件付き操作にも役立ちます。

「動的ドリルダウンの例」を参照してください。

<drilldown> エレメントによる条件付き操作用トークンの定義

条件による操作には、以下の事項が含まれています。

- 条件に基づいてトークン値を設定する。
- 視覚エフェクトの複数値フィールド用の値を選択する。
  複数値フィールドは、異なる値で複数回登場するフィールドです。
- トークン値に基づいて、開くビューを選択する。
- 条件に基づいて、パネルを非表示/表示する。

条件付き操作は、フォーム入力と条件付きドリルダウンの両方で利用できます。条件付きドリルダウンのトークン定義では、以下のタグをさまざまな組み合わせで使用します。

エレメント	説明
<drilldown>	ダッシュボードまたはフォーム内のフィールドの、リンク先を定義します。 <condition> を使って、カスタム・アクション用のトークンを設定することもできます。
<condition>	ドリルダウン・アクションの範囲を、特定のフィールドに制限します。
<selection>	<set> エレメントと一緒に使って、グラフのパン/ズーム機能の時間ウィンドウを設定します。 タイプが area、column、または line のグラフに適用します。 『シンプル XML リファレンス』の「グラフ・コントロール」および <selection> のエントリを参照してください。
<link>	ドリルダウンのリンク先を指定します。
<set>	トークン用の各種値を定義します。

<unset>	前に設定されたトークンを削除します。 設定されているトークンに基づく条件付き操作と一緒に使用します。

<set> エレメントを使ったトークンの定義

条件を利用するトークンを定義するには、<set> エレメントを使用します。<set> エレメントでトークンを定義する場合、他のトークンの値を利用することができます。sourcetype_tok トークンを定義するコード・スニペットの例を以下に示します。このトークンは、フィールド sourcetype の <table> エレメントからクリックされた値を取得します。

```
<drilldown>
  <condition field="sourcetype">
    <set token="sourcetype_tok">$click.value2$</set>
  </condition>
</drilldown>
```

サーチ内で sourcetype_tok トークンを使用することができます。

```
index=_internal sourcetype=$sourcetype_tok$ | timechart count by sourcetype
```

<condition> エレメントを使った視覚エフェクト内の複数値フィールドの値の選択

複数値フィールドは、イベント内に複数回登場するフィールドで、それぞれが異なる値を保有しています。詳細は、『ナレッジ管理マニュアル』の「複数値フィールドの設定」を参照してください。

複数値フィールドを表示するダッシュボードがある場合、<condition> エレメントを使ってクリックされたフィールドの値に対応するドリルダウン先を指定できます。フィールドの値に基づいて異なる宛先にリンクする例を以下に示します。<link> エレメントは、各条件に対して異なる事前定義トークンを使用します。例については、「複数値フィールドへのダッシュボードリンクの例」を参照してください。

```
<drilldown>
   <condition field="badges">
     <link >
       /app/foursquare_vegas/vegas_badge_1?form.badge=$click.value2$
     </link>
   </condition>

   <condition field="venue">
     <link>
       /app/foursquare_vegas/vegas_venue_1?form.venue=$row.venue$
     </link>
   </condition>

   <condition field="links">
    <link>
      http://www.yelp.com/search?find_desc=$row.venue$&find_loc=Las+Vegas,+NV
    </link>
   </condition>
</drilldown>
```

パン/ズーム・グラフ・コントロール用トークンの定義

Splunk Enterprise は事前定義トークンを使って、グラフのズーム機能を実装しています。ズーム機能を使って、グラフ内のデータシリーズの部分を選択し、個別のグラフに表示することができます。「パンとズームによるグラフ・コントロール」を参照してください。

グラフの子エレメントである <selection> エレメント内に、事前定義トークンの値を設定します。オリジナルのグラフでトークン値を使って、選択項目をズームする新たなグラフを表示します。

トークン	説明
start end	グラフ内の選択項目の、X 軸の開始/終了値を取得します。 グラフでのみ有効になります。ダッシュボード内の値にアクセスするために、定義したトークンに値を割り当てます。
start.<field> end.<field>	選択項目の Y 軸値を取得します。<field> は、グラフに表示されるシリーズを表しています。 グラフでのみ有効になります。ダッシュボード内の値にアクセスするために、定義したトークンに値を割り当てます。

時間グラフ内の選択項目のズーム例については、「パンおよびズームによるグラフ・コントロール」を参照してください。

トークンを使用するための構文

「トークン使用の概要」で説明したように、トークン値にアクセスするには、$...$ 区切り文字を使用します。たとえば、以下の視覚エフェクト用サーチは、field_tok トークンにアクセスします。フォーム入力は、以前に field_tok トークンを定義しています。

```
index=_internal source=*splunkd.log | stats count by $field_tok$
```

トークンのフィルタ

トークン・フィルタにより、トークンの値を正しく取得することができます。

フィルタ	説明
値を引用符で囲む $token_name\|s$	トークンが参照する値が引用符で囲まれるようにします。引用符内の値のすべての引用符文字 " がエスケープ処理されます。
HTML フォーマット $token_name\|h$	トークン値を HTML フォーマットとして有効にします。 <HTML> エレメントのトークン値は、デフォルトでこのフィルタを使用します。
URL フォーマット $token_name\|u$	トークン値を URL として有効にします。 <link> エレメントのトークン値は、デフォルトでこのフィルタを使用します。

トークンから返された値を引用符で囲むために、|s フィルタを使用したコード・スニペットを以下に示します。

```
<search>
  <query>
    index=_internal sourcetype=$sourcetype_tok|s$ | timechart count by sourcetype
   </query>
</search>
```

sourcetype_tok の値が *access_combined* の場合、以下のサーチ文字列が作成されます。

```
index=_internal sourcetype="access_combined" | timechart count by sourcetype
```

$ トークン区切り文字のエスケープ処理

$ 文字を含む静的なテキストを含める場合は、$$ を使ってトークン区切り文字値をエスケープ処理します。

リテラル値とトークン値の組み合わせ

トークンから返された値とリテラル値を組み合わせることができます。<set> エレメントを使用して、トークン値に基づく条件アクションを設定できます。

事前定義トークンから取得した値 click.value と静的なテキストを組み合わせたテンプレートを以下に示します。NewToken の値は引用符で囲まれます。

```
<set token="NewToken">sourcetype=$click.value|s$</set>
```

click.value の値が *access_combined* の場合、NewToken の値は以下のサーチ・フラグメントになります。

```
      sourcetype="access_combined"
```

<set> エレメントで prefix および suffix 属性を使用して、トークン値の静的なテキストを指定できます。NewToken の値を設定する例を以下に示します。これは、テンプレートの例と同じです。

```
<set token="NewToken" prefix="sourcetype=&quot;" suffix="&quot;">
  $click.value$
</set>
```

ユーザー・インターフェイス・コンポーネントを表示/非表示にするためのトークンへのアクセス

トークン値を使って、条件に応じてユーザー・インターフェイス・コンポーネントを表示/非表示にすることができます。以下のエレメントには、属性 depends および rejects が含まれています。これらの属性が使用するトークン値を設定するには、<set> および <unset> エレメントを使用します。

エレメント
<input>
<row>
<panel>
<chart>
<event>

<html>
<map>
<single>
<table>

たとえば、<chart> エレメントを `showChart` トークンが設定されている場合にのみ表示します。

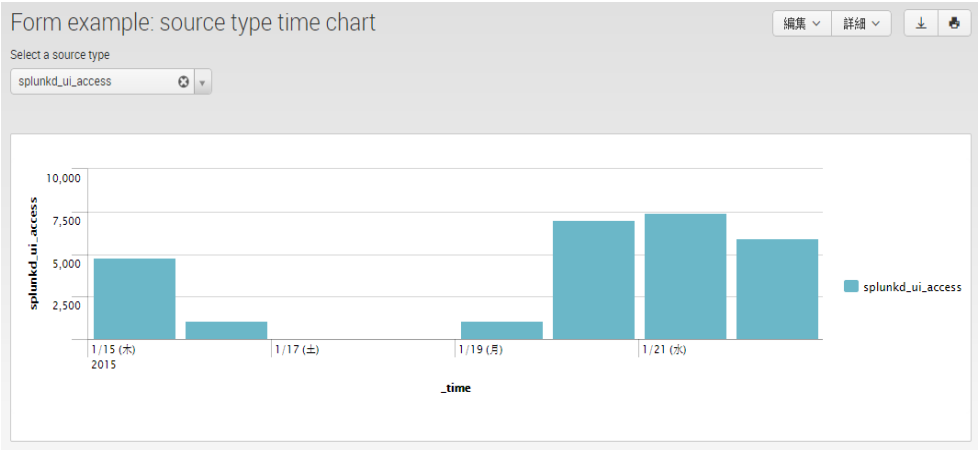
```
<chart depends="$showChart$">
```

トークンの使用例

フォーム入力の例

この例は、フォーム入力での基本的な使用方法を表しています。ドロップダウン・リストを使って、時間グラフのソースタイプを選択しています。「フォーム入力のトークンの定義」を参照してください。

<input> エレメントは、視覚エフェクトのサーチが使用する `sourcetype_tok` を定義しています。



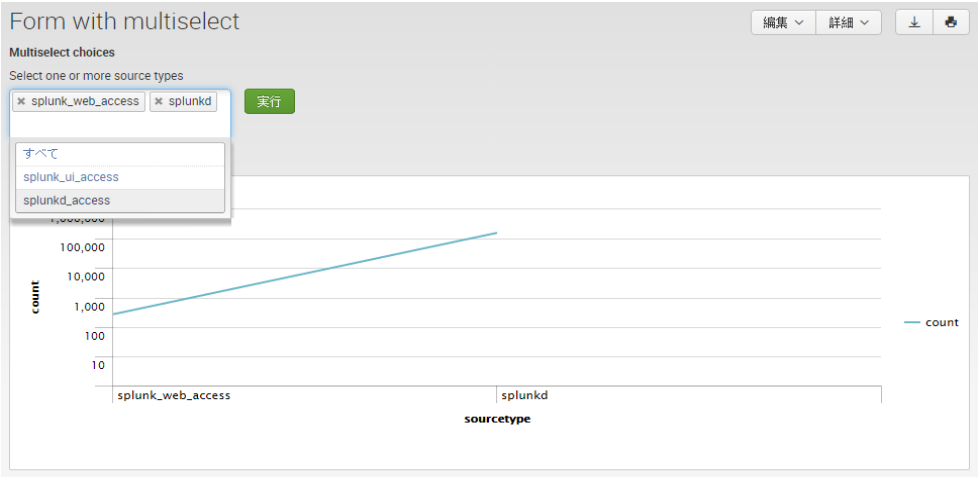

```
<form>
  <label>Form example: source type time chart</label>
  <fieldset autorun="true" submitButton="false">
    <input type="dropdown" token="sourcetype_tok">
      <label>Select a source type</label>
      <default>splunkd</default>
      <choice value="splunkd">splunkd</choice>
      <choice value="splunk_web_access">splunk_web_access</choice>
      <choice value="splunkd_ui_access">splunkd_ui_access</choice>
    </input>
  </fieldset>
  <row>
    <panel>
      <chart>
        <search>
          <query>
            index = _internal sourcetype=$sourcetype_tok$
            | timechart count by sourcetype
          </query>
            <earliest>-7d</earliest>
            <latest>-0d</latest>
        </search>
      </chart>
    </panel>
  </row>
</form>
```

複数選択入力の例

この例は、静的テキストとトークン値を使った、フォーム入力用サーチ文字列の作成方法を表しています。これは、複数選択オプションの作成に役立ちます。「複数選択入力のトークンの定義」を参照してください。

この例では <prefix>、<suffix>、<valuePrefix>、<valueSuffix>、および <delimiter> エレメントを使って、複数選択サーチ文字列を作成しています。ユーザーが splunkd および splunk_web_access を選択すると、以下のサーチ文字列が生成されます。

```
(sourcetype ="splunkd" OR sourcetype ="splunk_web_access")
```

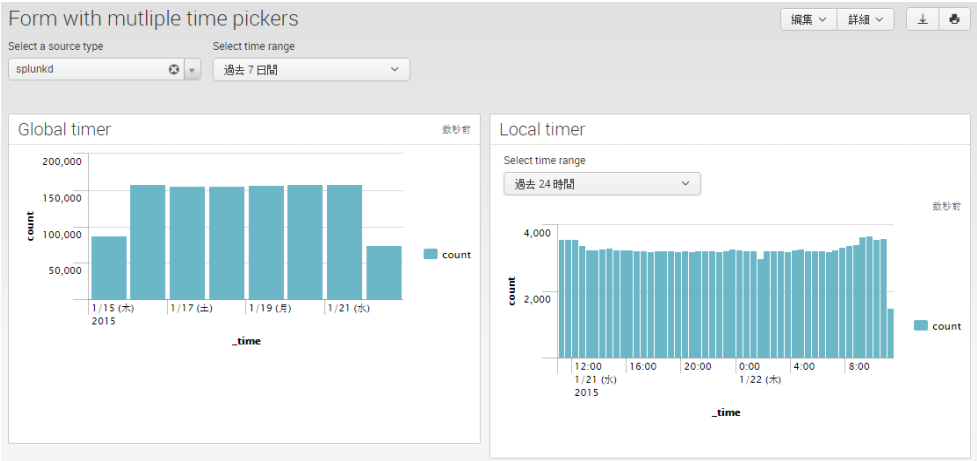



```
<form>
  <label>Form with multiselect</label>
  <fieldset autoRun="false" submitButton="true">
    <html>
      <p>
        <strong>Multiselect choices</strong>
      </p>
    </html>
    <input type="multiselect" token="sourcetype_tok" searchWhenChanged="false">
      <label>Select one or more source types</label>
      <choice value="*">All</choice>
      <choice value="splunk_web_access">splunk_web_access</choice>
      <choice value="splunkd">splunkd</choice>
      <choice value="splunk_ui_access">splunk_ui_access</choice>
      <choice value="splunkd_access">splunkd_access</choice>

      <!--      Build multiselect search:
        (sourcetype ="value1" OR sourcetype ="value2" OR ...)
      -->
      <prefix>(</prefix>
      <valuePrefix>sourcetype ="</valuePrefix>
      <valueSuffix>"</valueSuffix>
      <delimiter> OR </delimiter>
      <suffix>)</suffix>

    </input>
  </fieldset>
  <row>
    <panel>
      <title></title>
      <chart>
        <search>
          <query>index =_internal $sourcetype_tok$ | stats count by sourcetype</query>
          <earliest>-24h</earliest>
          <latest>now</latest>
        </search>
        <option name="charting.chart">line</option>
        <option name="charting.axisY.scale">log</option>
      </chart>
    </panel>
  </row>
</form>
```

時間入力の例

この例は、フォーム内でグローバル・タイム・ピッカーとローカル・タイム・ピッカーの両方を使用する方法を表しています。また、事前定義修飾子による時間入力トークンへのアクセス方法も表しています。「時間入力のトークンの定義」を参照してください。

この例は、グローバル・タイム・ピッカーとローカル・タイム・ピッカーの両方を持つフォームを表しています。<chart> エレメントにはローカル・タイム・ピッカーが含まれており、`local_time_input_tok` トークンへの修飾子を使ってもっとも早い時間ともっとも遅い時間の値にアクセスしています。



<form>

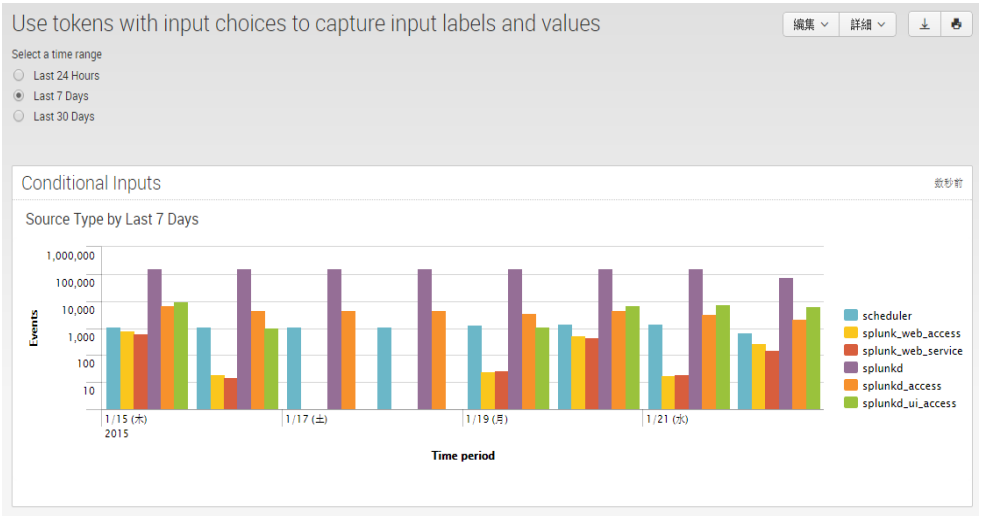
```
<label>Form with mutliple time pickers</label>
 <description></description>
 <fieldset submitButton="false">
   <input type="dropdown" token="source_tok" searchWhenChanged="true">
     <label>Select a source type</label>
     <choice value="*">All</choice>
     <search>
       <query>index=_internal | stats count by sourcetype</query>
       <earliest>-7d@h</earliest>
       <latest>now</latest>
     </search>
     <fieldForLabel>sourcetype</fieldForLabel>
     <fieldForValue>sourcetype</fieldForValue>
     <prefix>sourcetype="</prefix>
     <suffix>"</suffix>
     <default>splunkd</default>
   </input>

   <!-- Do not define token for global timer -->
   <input type="time" searchWhenChanged="true">
     <label>Select time range</label>
     <default>
       <earliest>-7d@h</earliest>
       <latest>now</latest>
     </default>
   </input>
 </fieldset>
 <row>
   <panel>
     <title>Global timer</title>
     <chart>
       <search>
         <query>index=_internal $source_tok$ | timechart count</query>
       </search>
     </chart>
   </panel>

   <panel>
     <title>Local timer</title>
    <!-- Define token for local timer -->
     <input type="time" searchWhenChanged="true" token="local_time_input_tok">
       <label>Select time range</label>
       <default>
         <earliest>-24h@h</earliest>
         <latest>now</latest>
       </default>
     </input>
    <chart>
       <search>
         <query>
           index=_internal $source_tok$ | timechart count
         </query>

         <!-- Use modifiers to token for a timer -->
         <earliest>$local_time_input_tok.earliest$</earliest>
         <latest>$local_time_input_tok.latest$</latest>
       </search>
     </chart>
   </panel>
 </row>
```

</form>

フォーム入力を使った条件操作

この例は、フォーム入力による条件操作の使用方法を表しています。「フォーム入力による条件付き操作用トークンの定義」を参照してください。

この例は、<change>、<condition>、および <set> エレメントを使って、選択された時間のラベルを条件に応じて設定し、もっとも早い時間のトークンを設定します。このサーチは、もっとも早い時間のトークンを使用して、サーチの時間境界を設定しています。この例では、入力選択項目に `label` および `value` 事前定義トークンを使用しています。「フォーム入力のラベルと値にアクセスする事前定義トークン」を参照してください。



注意：時間入力を除くすべての入力エレメントには、トークン属性が存在している必要があります。この例で、入力エレメントはトークン `period_tok` を定義しています。ただし、このトークンがサーチで使用されることはありません。

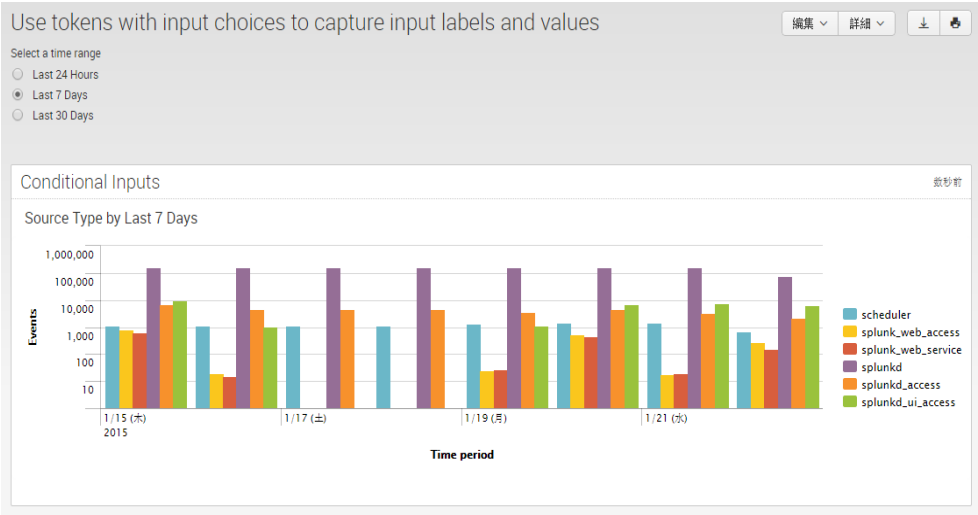
```
<form>
  <label>Use tokens with input choices to capture input labels and values</label>
  <fieldset submitButton="false">
    <input type="radio" token="period_tok">
      <label>Select a time range</label>
      <choice value="-24h@h">Last 24 Hours</choice>
      <choice value="-7d@d">Last 7 Days</choice>
      <choice value="-30d@d">Last 30 Days</choice>
      <default>Last 24 Hours</default>

      <change>
          <!-- use predefined input tokens to set -->
          <!-- tokens for the selected label and value -->
          <set token="date_label">$label$</set>
          <set token="earliest_tok">$value$</set>
      </change>

    </input>
  </fieldset>

  <row>
    <panel>
      <title>Conditional Inputs</title>
      <chart>
        <!-- Display selected label in the title -->
        <title>Source Type by $date_label$</title>

        <search>
          <query>index = _internal | timechart count by sourcetype</query>
          <!-- use the value of earliest_tok -->
          <!-- to set the time range         -->
          <earliest>$earliest_tok$</earliest>
          <latest>now</latest>
        </search>

        <option name="charting.axisY.scale">log</option>
        <option name="charting.axisTitleX.text">Time period</option>
        <option name="charting.axisTitleY.text">Events</option>
      </chart>
    </panel>
  </row>
</form>
```

フォーム入力のラベルと値へのアクセス

この例は、トークンを使ったフォーム入力のラベルと値へのアクセス方法を表しています。「フォーム入力のラベルと値にアクセスする事前定義トークン」を参照してください。

この例は、視覚エフェクトのタイトルにある、選択されたラジオ・ボタンのラベルを使用しています。選択された

ラジオ・ボタンの値を使って、サーチ範囲を判断しています。




```
<form>
  <label>Use tokens with input choices to capture input labels and values</label>
  <fieldset submitButton="false">
    <input type="radio" token="period_tok">
      <label>Select a time range</label>
      <choice value="-24h@h">Last 24 Hours</choice>
      <choice value="-7d@d">Last 7 Days</choice>
      <choice value="-30d@d">Last 30 Days</choice>
      <default>Last 24 Hours</default>

      <change>
          <!-- use predefined input tokens to set -->
          <!-- tokens for the selected label and value -->
          <set token="date_label">$label$</set>
          <set token="earliest_tok">$value$</set>
      </change>

    </input>
  </fieldset>

  <row>
    <panel>
      <title>Conditional Inputs</title>
      <chart>
        <!-- Display selected label in the title -->
        <title>Source Type by $date_label$</title>

        <search>
          <query>index = _internal | timechart count by sourcetype</query>
          <!-- use the value of earliest_tok -->
          <!-- to set the time range         -->
          <earliest>$earliest_tok$</earliest>
          <latest>now</latest>
        </search>

        <option name="charting.axisY.scale">log</option>
        <option name="charting.axisTitleX.text">Time period</option>
        <option name="charting.axisTitleY.text">Events</option>
      </chart>
    </panel>
  </row>
</form>
```

シンプル XML のカスタマイズ

シンプル XML ダッシュボードをカスタマイズして、ダッシュボードやフォームのレイアウトの変更、視覚エフェクトの追加、動作の変更を行うには、さまざまな方法があります。ここでは、カスタム CSS スタイルシートと JavaScript の使用方法、および一般的な JavaScript API の概要について説明していきます。

また、ダッシュボードのカスタマイズ例もいくつか取り上げます。

Splunk Apps の Splunk 6 Dashboard Examples App には、ダッシュボードの各種カスタマイズ方法が収録されています。

CSS と JavaScript

Splunk App は、ロード時に以下のファイルを参照します。これらのファイルは、App コンテキスト内のダッシュボードのデフォルトのスタイルと動作を定義しています。

- dashboard.css
- dashhboard.js

これらのファイルは、App のディレクトリ構造内に保管されています。

```
$SPLUNK_HOME/etc/apps/<app_name>/appserver/static/
```

これらのファイルを変更して、App 用の独自の機能を実装することができます。ただし、デフォルトのファイルに関する以下の事項に注意してください。

- ここで行うカスタマイズは、App 内の各ダッシュボードに適用されます。
- App の更新時に、カスタマイズしたファイルが上書きされる可能性があります。

独自のカスタム CSS および JavaScript ファイルを作成し、それをデフォルトファイルと同じディレクトリに保存することをお勧めします。次に、シンプル XML の <dashboard> または <form> エレメントで、それらのファイルを参照してください。

たとえば、<dashboard> エレメントから CSS および JavaScript ファイルを参照するには、以下のように指定します。

```
<dashboard stylesheet="myStyleSheet.css" script="myJavaScript.js">
  . . .
```

一般的な JavaScript API

SplunkJS スタックライブラリを参照する JavaScript コードを記述して、ダッシュボードの動作と視覚エフェクトをカスタマイズすることができます。シンプル XML ダッシュボード内で SplunkJS スタックを使用するには、スタックのフレームワークを理解しておくことが必要不可欠です。

> **注意**：SplunkJS スタックの説明はこのマニュアルの対象外です。SplunkJS スタックライブラリを使ったダッシュボードのカスタマイズの詳細は、Splunk 開発者向けポータルの「SplunkJS Stack」を参照してください。

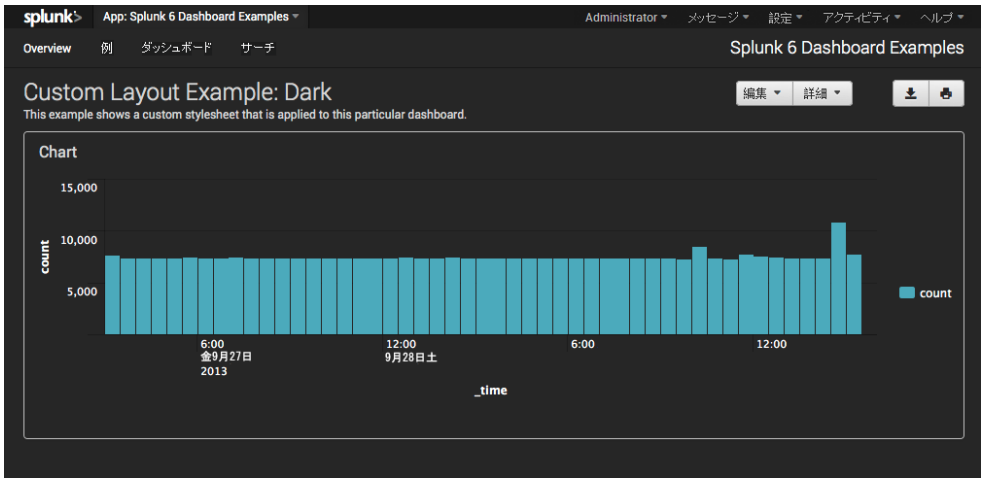
シンプル XML ダッシュボードをカスタマイズするための、JavaScript API は以下のカテゴリに分類できます。

- ダッシュボードレベル API
- 視覚エフェクトエレメント API
- カスタムテーブルセル表示 API

カスタマイズの例

カスタムスタイルシート

この例は、カスタムスタイルシートを使ってダッシュボードの外観を変更しています。



シンプル XML ソース

ダッシュボードエレメントはカスタムスタイルシート dark.css を参照しています。

$SPLUNK_HOME/etc/apps/myApp/appserver/static/dark.css

```
<dashboard stylesheet="dark.css">
  ...
```

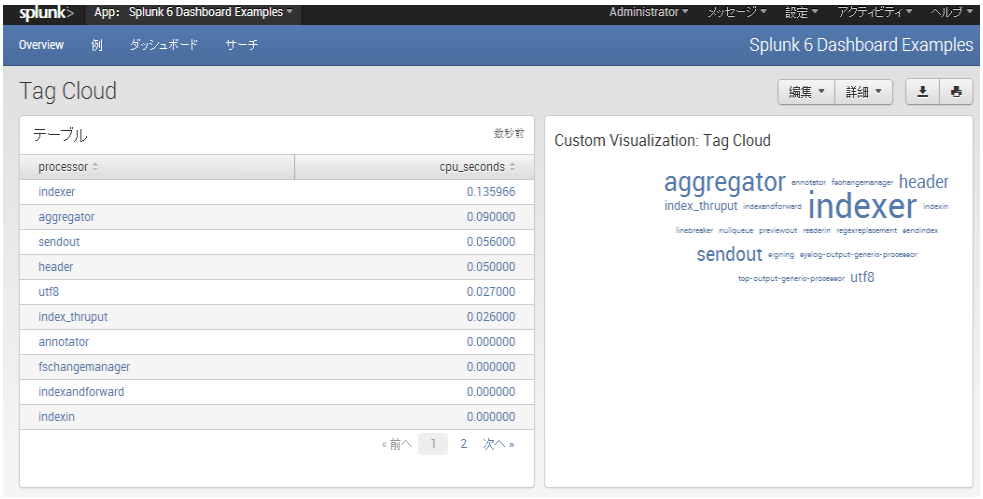
カスタムスタイルシート

dark.css の一部を以下に示します。完全なリストについては、「Splunk 6 Dashboard Examples」をダウンロードしてください。

```
body,.dashboard-body,.footer,.header,.app-bar,
.dashboard-panel,.dashboard-cell {
background: #333 !important;
}
a {
color: #ccc;
}
a:hover {
color: #fff;
}
.dashboard-row .dashboard-panel .panel-head h3,
.dashboard-header h2, p.description, .nav-footer>li>a {
color: #ddd;
text-shadow: none;
}
. . .
```

カスタム視覚エフェクト

この例は、サーチ結果のカスタム視覚エフェクトを定義しています。



このカスタマイズの主要コンポーネント：

- **シンプル XML ソース**
  <dashboard> エレメントは JavaScript ファイル custom_viz.js を参照します。

  <html> パネルを使って、その ID で参照されるカスタム視覚エフェクトを含めます。

- **custom_viz.js**
  SearchManager のインスタンスを作成して、視覚エフェクトのサーチを指定します。

  tagcloud.js で作成された TagCloud 視覚エフェクトのインスタンスを作成します。

- **tagcloud.js**
  サーチ結果に基づいて、タグクラウド視覚エフェクトを作成します。ドリル ダウン動作を実装します。

- **tagcloud.css**
  tagcloud.js のレイアウトを指定します。

グラフのカスタマイズ

ダッシュボードや他のビューにあるグラフは、柔軟にカスタマイズすることができます。Splunk Web のパネルエディタを使って、さまざまなカスタマイズを行えます。パネルエディタから、グラフの軸ラベルの変更、ゲージの色範囲の定義、縦棒/横棒グラフのスタックモードの設定、その他さまざまな作業を行えます。

パネルエディタが提供する基本カスタマイズオプションでは不十分な場合は、その内部にある XML を直接編集してグラフの外観や動作を変更することも可能です。軸ラベルのテキストの変更、グラフの軸の反転、グラフ結果に使用する特定のカラーパレットの定義、その他さまざまなカスタマイズ作業を行えます。

ここでは、シンプル XML を使った一般的なカスタマイズ作業の例を説明していきます。シンプル XML でのグラフの作成方法の詳細は、「視覚エフェクトの編集」を参照してください。利用できるグラフのカスタマイズオプ

ションの全一覧については、このマニュアルの「カスタムグラフ設定リファレンス」を参照してください。

グラフの色

グラフ内の**シリーズ**に対して、特定の色セットを指定することができます。グラフの色を変更するには、`seriesColors` プロパティ **charting.seriesColors** を使用します。`seriesColors` プロパティは、グラフ内のシリーズの色を調整するための、もっとも手軽な手段です。16進色値のリスト、0xRRGGBB の値をカンマで区切って指定し、それを角括弧で囲むことで、一連の色を表します。

シンプル XML ダッシュボードでの例：

```
<dashboard>
 <label>My dashboard</label>
 <row>
  <panel>
    <chart>
     <searchName>My saved report</searchName>
       <option name="charting.seriesColors">
         [0xFF0000,0xFFFF00,0x00FF00]
       </option>
     </chart>
    </panel>
  </row>
</dashboard>
```

グラフのシリーズの色はインデックスベースとなっており、凡例ラベルに割り当てられている他のすべてのシリーズにまたがって、それらのインデックスに基づいて特定の順序で、特定のシリーズに色が割り当てられます。(ここで使用している「インデックス」と言う言葉は、Splunk Enterprise のイベントインデックスやインデクサーのことではありません。)上で定義されている `seriesColors` リストで、グラフの最初のシリーズには色 `0xFF0000` (赤) が、2 番目のシリーズには色 `0xFFFF00` (黄) が、そしてそれ以降も同様に割り当てられていきます。

シンプル XML ダッシュボードでの例：

```
<dashboard>
 <label>My dashboard</label>
 <row>
  <panel>
    <chart>
     <searchName>My saved report</searchName>
      <option name="charting.legend.labels">[error,warn,info]</option>
      <option name="charting.seriesColors">[0xFF0000,0xFFFF00,0x00FF00]</option>
     </chart>
  </panel>
  </row>
</dashboard>
```

これにより、上記の `seriesColors` リストから、シリーズ `error` には色 `0xFF0000` (赤) が、シリーズ `warn` には色 `0xFFFF00` (黄) が、それ以降も同様に色が割り当てられていきます。

ただし、ビュー内にその他のグラフも存在している場合、この対応付けが必ずしも保証される訳ではありません。デフォルトでは、ビュー内のすべてのグラフのすべての凡例は、色を同期するために共通の「マスター凡例」に接続しています。マスター凡例が、その「スレーブ」凡例の結合されたマッピングから、最終的なラベル/シリーズのインデックスマッピングを決定します。マスター凡例が処理する最初のスレーブ凡例 (一般的にはビュー内の最初の項目) のラベルは、次に処理される凡例のラベルの前に配置されています (重複項目を差し引く)。そのため、上のラベルリスト内のシリーズインデックス 0 の `error` が、マスターリストのシリーズインデックス 0 に必ずしも存在する訳ではありません。この問題に対処するために、凡例の `masterLegend` に NULL または空文字列を割り当てることで、凡例を同期化から除外することができます。

シンプル XML ダッシュボードでの例：

```
<dashboard>
 <label>My dashboard</label>
 <row>
  <panel>
    <chart>
      <searchName>My saved report</searchName>
      <option name="charting.legend.labels">[error,warn,ok]</option>
      <option name="charting.seriesColors">[0xFF0000,0xFFFF00,0x00FF00]</option>
      <option name="charting.legend.masterLegend"></option>
     </chart>
  </panel>
  </row>
</dashboard>
```

こうすることにより、上記の labels および seriesColors リスト間での 1 対 1 のマッピングが保証されます。

シリーズインデックスではなく名前に基づいて特定のシリーズに色を割り当てる場合はどうしたら良いでしょうか？このような場合は、fieldColors プロパティを使って、特定の色と特定のフィールドをマッピングします。fieldColors プロパティは、マップするフィールド/色のペアを波括弧で囲む形で表されます。

```
<option name="charting.fieldColors">
  {"error":0xFF0000,"warn":0xFFFF00,"info":0x00FF00}
</option>
```

この例では、シリーズ error に色 0xFF0000 (赤)、シリーズ warn に色 0xFFFF00 (黄)、シリーズ info に色 0x00FF00 (緑) を割り当てます。この例では必要ありませんが、文字 []{}(),:" のいずれかを含むフィールド名は、二重引用符で囲む必要があります。フィールド名内の既存の二重引用符または円記号の前には、円記号を付けてエスケープ処理する必要があります。

ブラシとパレット

シリーズの「色」の概念は、グラフ内のシリーズや他の視覚的要素のスタイルを大幅に簡素化することにあります。たとえば、Splunk グラフ内のすべてのシリーズのデフォルトスタイルは、単色ではなく 2 つの色の階調になっています。前述した seriesColors プロパティは、利便性に優れており、グラフのスタイル定義の複雑製を大幅に簡素化します。グラフのデフォルトのシリーズ色マッピングを変更することにのみ興味があり、その他のスタイルはデフォルト設定で構わない場合は、seriesColors プロパティでも十分に役立ちます。しかし、デフォルトの階調や色を超えた入念なスタイル設定を行いたい場合は、その基盤となるブラシとパレットアーキテクチャを理解する必要があります。

テキストを除く Splunk グラフのすべての視覚的要素は、ブラシを使って「ペイント」されています。ブラシは、塗りつぶし、ストローク、階調、画像、そしてビデオでさえもペイントすることができます。一部のブラシはこれらの方法を組み合わせたり、レイヤー化したりすることも可能です。たとえば、Solid Fill Brush は単色の塗りつぶしをペイントします。Solid Stroke Brush は、単色のストロークをペイントします。また、複数のブラシで同時にペイントする、Group Brush も用意されています。たとえば Group Brush を使って、単色ストロークに囲まれた単色塗りつぶしをペイントすることができます。

50% の透明度の赤の単色塗りつぶしを行うカスタムブラシの定義例を以下に示します。

```
<dashboard>
 <label>My dashboard</label>
 <row>
  <chart>
   <searchName>My saved report</searchName>
    <option name="charting.myBrush">solidFill</option>
    <option name="charting.myBrush.color">0xFF0000</option>
    <option name="charting.myBrush.alpha">0.5</option>
  </chart>
 </row>
</dashboard>
```

グラフには、しばしば描画する複数のシリーズが存在しています。この場合、一般的には、各シリーズを表すために複数のブラシが必要です。しかし、異なるグラフ視覚エフェクトオプションすべてに対して、個別のシリーズ用の独自のブラシを設計することに時間を費やすことは容易ではありません (特に多数のシリーズを持つビューがある場合)。代わりに、グラフはブラシの*パレット*を使用しています。ブラシのパレットは、シリーズインデックスをブラシにマップしています。Splunk のグラフ用にさまざまなブラシパレットが用意されています。一番簡単なブラシパレットは Solid Fill Brush パレットで、グラフ内の各シリーズに対して単色の塗り潰しブラシを生成します。

各ブラシが生成する色を決定するために、Solid Fill Brush パレットは他の種類のパレット、カラーパレットを使用します。ブラシパレットと同様に、カラーパレットはシリーズインデックスを色にマップしています。たとえば、List Color Palette はシリーズインデックスを、指定した色リストからの色に直接マップします。デフォルトでは、インデックスが色リストの範囲外になる場合、リストの先頭に戻って色が繰り返し使用されます。必要に応じて List Color Palette は、色を再利用する代わりに複数の色リストで補完して、シリーズの合計数にまたがる色範囲を生成することができます。以下の例は、赤、緑、青間で補完するカラーパレットを表しています。

```
<dashboard>
 <label>My dashboard</label>
 <row>
  <chart>
   <searchName>My saved report</searchName>
    <option name="charting.myColorPalette">list</option>
    <option name="charting.myColorPalette.colors">[0xFF0000,0x00FF00,0x0000FF]</option>
    <option name="charting.myColorPalette.interpolate">true</option>
   </chart>
  </row>
</dashboard>
```

プロパティ参照

上記で定義したカラー・パレットを使ってグラフの Solid Fill Brushes を生成するには、Solid Fill Brush Palette の適切なプロパティからそれを参照します。他のプロパティの値として使用するプロパティを参照するには、「@」記号に続けて参照するプロパティ名を指定します (プリフィックス「charting」を差し引く)。Solid Fill Brush Palette には、値としてカラーパレットを予期する `colorPalette` プロパティがあります。

```
<option name="charting.myBrushPalette">solidFill</option>
<option name="charting.myBrushPalette.colorPalette">@myColorPalette</option>
```

再びプロパティ参照を使用して、myBrushPalette から生成されたブラシを使って縦棒をペイントする縦棒グラフを作成します。縦棒グラフには、この目的のために特別に設計された `columnBrushPalette` プロパティがあります。

```
<option name="charting.chart">column</option>
<option name="charting.chart.columnBrushPalette">@myBrushPalette</option>
```

また、元の myColorPalette 定義内のプロパティ参照を使って、色のリストを定義する他のプロパティを参照することもできます。そしてこのことが、前述の単純な seriesColors プロパティを、Splunk グラフのデフォルトのブラシ/パレットセットと関連付ける仕組みとなります。

```
<option name="charting.myColorPalette.colors">@seriesColors</option>
```

その場でのグラフオプションの作成

単にその場で宣言することで、独自のプロパティを即座に定義することができます。たとえば、以下の項目は独自の赤単色の塗り潰しブラシを宣言しています。

```
   <option name="charting.myBrush">solidFill</option>
   <option name="charting.myBrush.color">0xFF0000</option>
```

参照またはコピーを使って、あるプロパティを他のプロパティに割り当てることができます。「@」記号は、プロパティ参照を、「#」記号はプロパティコピー (クローン) を表します。たとえば、前述のブラシをツールヒントの背景に使用するために、以下のプロパティ参照を使用できます。

```
   <option name="charting.tooltip.backgroundBrush">@myBrush</option>
```

ただし、アルファ透明度が 50% であることを除いて、前述の定義されたカスタムブラシと同じブラシを使用したい場合は、それのクローンを作成して、それのアルファプロパティを変更することができます。

```
   <option name="charting.tooltip.backgroundBrush">#myBrush</option>
   <option name="charting.tooltip.backgroundBrush.alpha">0.5</option>
```

文字列の最初の文字として「@」または「#」を使用する必要がある場合 (たとえば、軸のタイトルとして)、エスケープ処理するために記号を 2 つ指定します。

```
   <option name="charting.axisTitleY.text">## Of Downloads</option>
   <option name="charting.axisTitleX.text">@@Foo</option>
```

シンプル XML ビューリファレンス

シンプル XML リファレンス

ダッシュボードとフォーム

dashboard

<dashboard>
ビューのルート・エレメント。ダッシュボードには 1 つまたは複数の行が含まれており、それぞれに 1 つまたは複数のパネルを表示できます。 ダッシュボードには、それに表示するデータを取得する 1 つまたは複数のグローバル <search> エレメントを含めることができます。<panel> エレメントには、各パネルのデータを取得する 1 つまたは複数の <search> エレメントを指定することができます。 ダッシュボードにグローバル・サーチが含まれている場合、<panel> エレメントにはサーチからデータを表示するための後処理サーチが必要です。
`<dashboard>`

```
<label> (0..1)
<description> (0..1)
<search> (0..1)
<row> (1..n)
  <panel> (0..n)
    <search> (0..n)
    <chart> |  <event> | <html> | <map> | <single> | <table> (1..n)
      <search> (0..n, for each visualization element)
```

属性

名前	タイプ	デフォルト	説明
hideChrome hideAppBar hideEdit hideFooter hideSplunkBar hideTitle	論理値	False	ダッシュボードから標準 Splunk Enterprise ダッシュボード・コンポーネントを削除するための属性。 **Chrome**：Splunk バー、App バー、およびフッターを非表示にします。 **App バー**：Splunk Enterprise アプリケーションとビューを記載したバー。 **編集**：ダッシュボードの編集を可能にする、ドロップダウン・リストとコンポーネント。有効にした場合、**[設定] > [ユーザー・インターフェイス] > [ビュー]** または [ダッシュボード] ページを使ってダッシュボードを編集します。 **フッター**：ダッシュボードのフッターに、Splunk Enterprise リンクと著作権情報を記載します。 **Splunk バー**：Splunk Enterprise ホーム・ページへのリンクと [設定] ページへのドロップダウン・リストを提供する上部バー。 **タイトル**：ダッシュボードの <label> および <description> エレメントに定義されているテキスト。
isDashboard	論理値	True	内部利用。 ビューがダッシュボードか、またはダッシュボードではないアドバンスト XML で作成されたビューかを表しています。
isVisible	論理値	True	ダッシュボードが App 内のダッシュボード一覧および App のナビゲーション・メニューに記載されているかどうかを示します。
onunloadCancelJobs	論理値		ユーザーがダッシュボードから別の場所に移動した時に、サーチ・ジョブをキャンセルするかどうかを示します。
refresh	整数	0	ダッシュボードの更新間隔を秒で指定します。指定した時間の経過後にダッシュボードが再ロードされます。
script	文字列		ロードするカスタム JavaScript ファイルのカンマ区切りリスト。ファイルは以下の場所にあります。ファイルをサブディレクトリに配置することはできません。 `$SPLUNK_HOME/etc/apps/<app_name>/appserver/static/` 他のApp からカスタム JavaScript ファイルを参照するには、ファイルを参照する際に App 名を指定します：例： <dashboard script="myApp:myScript.js">
stylesheet	テキスト		ダッシュボードで使用するスタイルシートを指定します。スタイルシートのファイルは以下の場所にあります。ファイルをサブディレクトリに配置することはできません。 `$SPLUNK_HOME/etc/apps/<app_name>/appserver/static/` 他のApp からカスタムのスタイルシートファイルを参照するには、ファイルを次のように参照する際に App 名を指定します： <dashboard script="myApp:myStyles.css">

例

```
<dashboard script="myScript.js, myScript2.js" stylesheet="myLocalStyles.css, myApp:myAppStyles.css">
  <label>Data inputs</label>
  <description>Listing of data inputs</description>
```

```
  <row>
    <panel>
      <chart>
        <title>Source types last 7 days</title>
        <search ref="Source types last 7 days report" />
      </chart>
    </panel>
  </row>
</dashboard>
```

form

<form>
ユーザー入力エレメントを持つダッシュボード。ユーザー入力エレメントは、フォーム内で使われているサーチ用の 1 つまたはす複数の用語値を提供します。

```
<form>
  <label> (0..1)
  <description> (0..1)
  <search> (0..1)
  <fieldset> (1)
    <input> (1..n)
  <row> (1..n)
    <panel> (0..n)
      <search> (0..n)
      <chart> |  <event> | <html> | <map> | <single> | <table> (1..n)
        <search> (0..n, for each visualization element)
```

属性

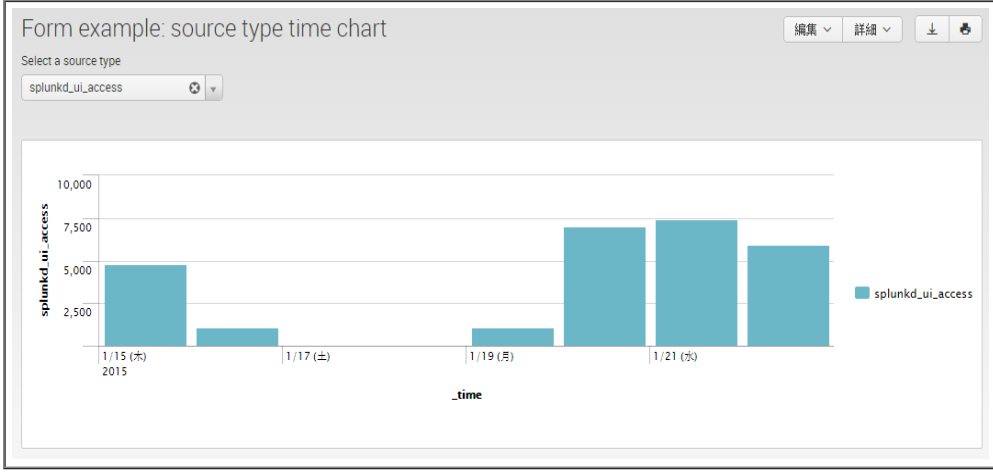
名前	タイプ	デフォルト	説明
hideChrome hideAppBar hideEdit hideFooter hideSplunkBar hideTitle	論理値	False	ダッシュボードから標準 Splunk Enterprise ダッシュボード・コンポーネントを削除するための属性。 **Chrome**：Splunk バー、App バー、およびフッターを非表示にします。 **App バー**：Splunk Enterprise アプリケーションとビューを記載したバー。 **編集**：ダッシュボードの編集を可能にする、ドロップダウン・リストと関連コンポーネント。有効にした場合、**[設定] > [ユーザー・インターフェイス] > [ビュー]** または [ダッシュボード] ページを使ってダッシュボードを編集します。 **フッター**：ダッシュボードのフッターに、Splunk Enterprise リンクと著作権情報を記載します。 **Splunk バー**：Splunk Enterprise ホーム・ページへのリンクと [設定] ページへのドロップダウン・リストを提供する上部バー。 **タイトル**：ダッシュボードの <label> および <description> エレメントに定義されているテキスト。
isDashboard	論理値	True	内部利用。 ビューがダッシュボードか、またはダッシュボードではないアドバンスト XML で作成されたビューかを表しています。
isVisible	論理値	True	ダッシュボードが App 内のダッシュボード一覧および App のナビゲーション・メニューに記載されているかどうかを示します。
onUnloadCancelJobs	論理値		ダッシュボードから別の場所に移動した時に、サーチジョブをキャンセルするかどうかを示します。
refresh	整数	0	更新間隔を秒で指定します。指定した更新間隔の経過後にダッシュボードが再ロードされます。 ロードするカスタム JavaScript ファイルのカンマ区切りリスト。ファイルは以下の場所にあります。ファイルをサブディレクトリに配置することはできません。

script	文字列	`$SPLUNK_HOME/etc/apps/<app_name>/appserver/static/`
		他のApp からカスタム JavaScript ファイルを参照するには、ファイルを次のように参照する際に App 名を指定します：
		`<form script="myApp:myScript.js">`

stylesheet	テキスト	フォームで使用するスタイルシートを指定します。スタイルシートのファイルは以下の場所にあります。ファイルをサブディレクトリに配置することはできません。
		`$SPLUNK_HOME/etc/apps/<app_name>/appserver/static/`
		他のApp からカスタムのスタイルシートファイルを参照するには、ファイルを次のように参照する際に App 名を指定します。ファイルをサブディレクトリに配置することはできません。
		`<form script="myApp:myStyles.css">`

例

```
<form script="myLocalScript.js, myApp:myAppScript.js" stylesheet="myStyles.css, myStyles2.css">
  <label>Form example: source type time chart</label>
  <fieldset autorun="true" submitButton="false">
    <input type="dropdown" token="sourcetype_tok">
      <label>Select a source type</label>
      <default>splunkd</default>
      <choice value="splunkd">splunkd</choice>
      <choice value="splunk_web_access">splunk_web_access</choice>
      <choice value="splunkd_ui_access">splunkd_ui_access</choice>
    </input>
  </fieldset>
  <row>
    <panel>
      <chart>
        <search>
          <query>
            index = _internal sourcetype=$sourcetype_tok$
            | timechart count by sourcetype
          </query>
            <earliest>-7d</earliest>
            <latest>-0d</latest>
        </search>
      </chart>
    </panel>
  </row>
</form>
```



panel

<panel>
1 つまたは複数のパネル視覚エフェクトエレメントを表示、グループ化するコンテナ。 複数の視覚エフェクト・エレメントは、パネル内に垂直にグループ化されます。例外は単一の視覚エフェクトエ

レメントです。複数の単一エレメントは、水平にグループ化されます。

インラインとリファレンスの 2 種類のパネルが存在しています。

インライン・パネル
インライン・パネルには、1 つまたは複数の視覚エフェクト・エレメントが含まれています。ダッシュボード・エディタやパネル・エディタを使って、インライン・パネルを作成、編集することができます。シンプル XML ソース・コードでパネルを編集することも可能です。

リファレンス・パネル
リファレンス・パネルは、ダッシュボードのプレビルト・パネルのコンテンツを表示します。リファレンス・パネルには、プレビルト・パネルへの参照を示す `ref` 属性と、必要に応じてオプションの `app` 属性が含まれています。

リファレンス・パネルは、ダッシュボードの XML コードに指定した <panel> の子エレメントを認識しません。

パネル・エディタを使ってリファレンス・パネルのコンテンツを編集することはできません。

プレビルト・パネルの詳細は、「ダッシュボードのパネル」および「参照によるパネルの作成と追加」を参照してください。

親エレメント

<row>

インライン・パネル

```
<row>
  <panel> (0..n)
    <title> (0..1)
    <description> (0..1)
    <search> (0..n)
    <chart> |  <event> | <html> | <map> | <single> | <table> (1..n)
```

リファレンス・パネル

```
<row>
  <panel ref="[panel name]" [app="[app name]"]> (0..n)
    <!-- Other <panel> child elements ignored -->
```

子エレメント (インライン・パネル)

エレメント	タイプ	デフォルト	説明
<description>	テキスト		パネルに表示する説明テキスト。
パネルの視覚化エレメント	テキスト		サーチの結果を表示する視覚化エレメント。 HTML によるテキストを表示する <html> パネルになることも可能です。「パネルの視覚化エレメント」を参照してください。
<search>	テキスト		サーチ文字列。 後処理サーチで利用できるベースサーチ。
<title>	テキスト		パネルのタイトルを表示します。

属性

名前	タイプ	デフォルト	説明
ref	テキスト		**(必須) リファレンス・パネルにのみ適用されます。** プレビルト・パネルの名前を参照します。これは、**[設定] > [ユーザー・インターフェイス] > [パネル]** に表示される名前です。
app	テキスト	説明を参照。	**(オプション) リファレンス・パネルにのみ適用されます。** リファレンス・パネルを含む App の名前を参照します。リファレンス・パネルの App は、**[設定] > [ユーザー・インターフェイス] > [パネル]** に表示されます。 `app` のデフォルト値は、ダッシュボードがある App です。
	トーク		

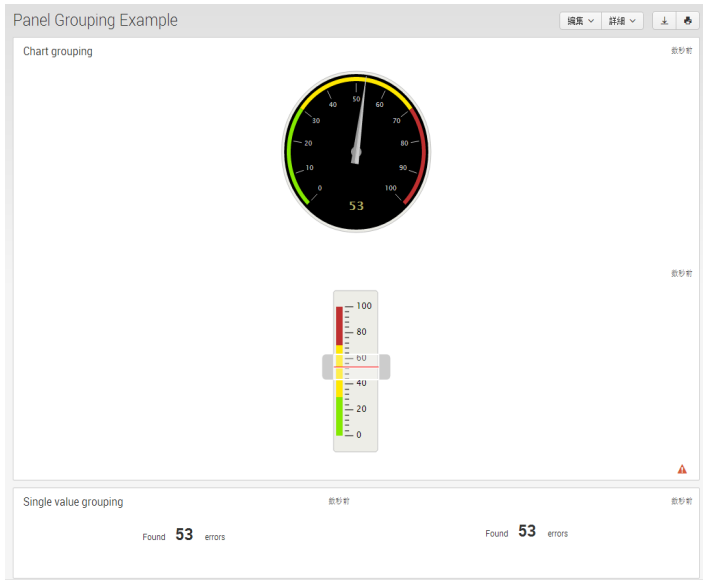
depends	トークンのカンマ区切りリスト	ダッシュボード内にこのパネルを表示するには、リストに記載されている 1 つまたは複数のトークンが存在している必要があります。
id	テキスト	パネルの ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。 以下の用語は内部使用向けに予約されており、`id` に使用することはできません。 dashboard, default, footer, header, search, submitted, url
rejects	トークンのカンマ区切りリスト	ダッシュボードへのこのパネルの表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。

例

<panel> エレメントを使って、グラフ視覚エフェクトと単一値視覚エフェクトをグループ化します。

```
<dashboard>
  <label>Dashboard Panel Example</label>
  <description></description>
  <row>
    <panel>
      <chart>
        <title>Chart grouping</title>
        <search>
          <query>
          index=_internal source="*splunkd.log"
          ( log_level=ERROR OR log_level=WARN*
          OR log_level=FATAL OR log_level=CRITICAL )
          | stats count as log_events
          | rangemap field=log_events low=1-100 elevated=101-300 default=severe
          </query>
          <earliest>-7d@h</earliest>
          <latest>now</latest>
        </search>
        <option name="charting.chart">radialGauge</option>
      </chart>
      <chart>
        <search>
          <query>
          index=_internal source="*splunkd.log"
          ( log_level=ERROR OR log_level=WARN*
          OR log_level=FATAL OR log_level=CRITICAL )
          | stats count as log_events
          | rangemap field=log_events low=1-100 elevated=101-300 default=severe
          </query>
          <earliest>-7d@h</earliest>
          <latest>now</latest>
        </search>
        <option name="charting.chart">markerGauge</option>
      </chart>
    </panel>
  </row>
  <row>
    <panel>
      <single>
        <title>Single value grouping</title>
         <search>
           <query>
          index=_internal source="*splunkd.log"
          ( log_level=ERROR OR log_level=WARN*
          OR log_level=FATAL OR log_level=CRITICAL )
```

```
          | stats count as log_events
          | rangemap field=log_events low=1-100 elevated=101-300 default=severe
           </query>
          <earliest>-7d@h</earliest>
          <latest>now</latest>
        </search>
        <option name="beforeLabel">Found</option>
        <option name="afterLabel">errors</option>
      </single>
      <single>
         <search>
           <query>
          index=_internal source="*splunkd.log"
          ( log_level=ERROR OR log_level=WARN*
          OR log_level=FATAL OR log_level=CRITICAL )
          | stats count as log_events
          | rangemap field=log_events low=1-100 elevated=101-300 default=severe
           </query>
          <earliest>-7d@h</earliest>
          <latest>now</latest>
        </search>
        <option name="beforeLabel">Found</option>
        <option name="afterLabel">errors</option>
      </single>
    </panel>
  </row>
</dashboard>
```



row

<row>
ダッシュボードやフォームに水平レイアウトで 1 つまたは複数の視覚エフェクトエレメントを表示するためのコンテナ。 行内の視覚化エレメントをグループ化するには、<panel> エレメントを使用します。
親エレメント <dashboard> | <form>
<row>   <panel> (0..n)

属性

名前	タイプ	デフォルト	説明

depends	トークンのカンマ区切りリスト		ダッシュボード内にこの行を表示するには、リストに記載されている 1 つまたは複数のトークンが存在している必要があります。1 つまたは複数のトークンを指定することができます。
grouping	整数のカンマ区切りリスト	グループ化なし	**非推奨。視覚化エレメントをグループ化するには、<panel> エレメントを使用します。** 1 行内のパネルのグループ化を、グループ化するパネル数を表す数字のカンマ区切りリストで設定します。パネルをグループ化すると、そのグループの各パネルの視覚エフェクトがコンテナ内に配置されます。例外的に、コンテナをパネル視覚エフェクト用の列とみなすことができます。視覚エフェクトは、コンテナ内に上下に配置されます。グループに <single> タイプの視覚エフェクトのみが含まれている場合、視覚エフェクトは横方向に並べて配置されます。 グループ化の最初の数字が、最初のグループの初期パネル数を表します。リスト内のそれ以降の数字は、次のパネル・セットのグループを形成します。 たとえば、6 つの視覚エフェクトを持つ行を考えてみましょう。以下のグループ化を指定します。 `<row grouping="2,1,3">` この場合、最初の 2 つのパネル用のコンテナ、1 つの視覚エフェクトを持つ 2 番目のコンテナ、および最後の 3 つのパネルをグループ化する 3 番目のコンテナが作成されます。
id	テキスト		行の ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。 以下の用語は内部使用向けに予約されており、`id` に使用することはできません。 dashboard, default, footer, header, search, submitted, url
rejects	トークンのカンマ区切りリスト		ダッシュボードへのこの行の表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。

例

<panel> エレメントの例を参照してください。この例は、<panel> エレメントを使った行内の視覚エフェクトのグループ化を表しています。

label

<label>
ダッシュボード、フォーム、またはフォーム入力のヘッダー・テキスト (省略可)。
親エレメント <dashboard> | <form>
`<label>[text]</label> (0..1)`

例
`<form>`
`  <label>Event count for different source types</label>`
`  . . .`
`  <fieldset>`
`  <input type="text" token="series">`

```
    <label>Enter a source type</label>
    <default></default>
    <seed>splunkd</seed>
  </input>
  </fieldset>
  . . .
</form>
```

説明

<説明>
<dashboard>、<form>、または <panel> 下に表示するテキスト。
親エレメント <dashboard> | <form> <panel>
`<description>[text]</description> (0..1)`

例
`<dashboard>`
`  <label>Event count for different source types</label>`
`  <description>Listing of common source types</description>`
`  . . .`
`  <panel>`
`  <title>Source types for the last 7 days</title>`
`  <description>Count for each source type in the internal index</description>`
`  . . .`
`  . . .`
`</dashboard>`

フォーム入力

fieldset

<fieldset>
フォームの入力エレメントを定義します。
親エレメント <form>

`<fieldset autoRun="[Boolean]" submitButton="[Boolean]">`
`  <html> (0..n)`
`  <input type="[input type]" token="[search token]"> (1..n)`
`  <default> (0..1)`
`  <fieldForLabel> (0..1)`
`  <fieldForValue> (0..1)`
`  <label> (0..1)`
`  <prefix> (0..1)`
`  <search> (0..1)`
`  <seed> (0..1)`
`  <selectFirstChoice> (0..1)`
`  <suffix> (0..1)`
`  <populatingSearch> | <populatingSavedSearch> (0..1, deprecated)`

属性

名前	タイプ	デフォルト	説明
autoRun	論理値	False	ページのロード時にサーチを実行するかどうかを示します。
submitButton	論理値	True	[送信] ボタンを表示するかどうかを示します。

例

```
<fieldset autoRun="true" submitButton="false">
  <input type="text" token="series">
    <label>sourcetype</label>
    <default></default>
    <seed>splunkd</seed>
    <suffix>*</suffix>
  </input>
</fieldset>
```

input (チェックボックス)

<input type="checkbox">

フォームのチェックボックス入力を定義します。

属性

名前	タイプ	デフォルト	説明
depends	トークンのカンマ切り区リスト		この入力を表示するには、リストに記載されている 1 つまたは複数のトークンが存在している必要があります。1 つまたは複数のトークンを指定することができます。フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
id	テキスト		この入力の ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。
rejects	トークンのカンマ切り区リスト		この入力の表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
searchWhenChanged	論理値		選択が変更された時にサーチを実行することを指定します。
token	文字列		指定された値に置換するサーチ文字列内のトークンを指定します。

親エレメント

<fieldset>

```
<input type="checkbox" token="[search token]"> (1..n)
  <default> (0..1)
  <delimiter> (0..1)
  <label> (0..1)
  <prefix> (0..1)
  <search> (0..1)
  <suffix> (0..1)
  <valuePrefix> (0..1)
  <valueSuffix> (0..1)
```

子エレメント

エレメント	タイプ	デフォルト	説明
<change>	<condition>		条件付きアクションを設定する、入力の選択項目を指定します。 複数選択入力に、<change> エレメントは使用できません。「<change>」を参照してください。
<condition>	入力選択項目		条件付きアクションを設定する、入力の選択項目 1 つを指定します。

			「<condition> (input)」を参照してください。
<default>	属性値		入力エレメントのデフォルト値を指定します。
<delimiter>	テキスト	説明を参照	選択した各値の間に配置する文字列です。一般的には、大文字を使って「 OR 」または「 AND 」を指定します。引用符は指定しないでください。また、テキストの前後には、スペース文字を指定してください。 **デフォルト値**：" " デフォルト値に引用符は含まれていません。この引用符は、デフォルト値がスペースであることを示すために、便宜的に用いられています。
<earliest> <latest>	テキスト		もっとも早い/もっとも遅い時間パラメータを示す時間式。入力の選択項目を動的に設定するには、<search> エレメントと一緒に使用します。 時間は相対時間または絶対時間で指定できます。相対時間の場合、相対時間修飾子を使用します。「サーチへの時間修飾子の指定」を参照してください。絶対時間の場合、時間は UNIX エポック時フォーマットで指定します。
<fieldForLabel> <fieldForValue>	テキスト		<search> エレメントを使って入力の選択項目を動的に設定する場合に、ラベルと値に対して使用するフィールド。
<label>	テキスト		入力エレメントで表示するテキスト。
<prefix>	テキスト		入力エレメントの値の先頭に付ける文字列。正規表現を使用できます。
<search>	テキスト		入力の選択項目を動的に設定するサーチ。レポートからのサーチを参照するには、<search> エレメントの `ref` 属性を使用します。「<search>」を参照してください。
<suffix>	テキスト		入力エレメントの値の最後に追加する文字列。正規表現を使用できます。
<valuePrefix>	テキスト	"	入力エレメントの値の先頭に付ける文字列。正規表現を使用できます。 デフォルト値は左二重引用符になります (")。
<valueSuffix>	テキスト	"	入力エレメントの値の最後に追加する文字列。正規表現を使用できます。 デフォルト値は右二重引用符になります (")。

例

この例は、ユーザーが複数選択から「One」と「Three」を選択した時に、以下の文字列を生成します。

("1*" AND "3*")

```
<fieldset>
  <input type="checkbox" token="mv5">
    <choice value="1">One</choice>
    <choice value="2">Two</choice>
    <choice value="3">Three</choice>
    <delimiter> AND </delimiter>
    <prefix>(</prefix>
    <suffix>)</suffix>
    <valuePrefix>"</valuePrefix>
    <valueSuffix>*"</valueSuffix>
  </input>
</fieldset>
```

入力 (ドロップダウン)

<input type="dropdown">
フォームのドロップダウン入力を定義します。

属性

名前	タイプ	デフォルト	説明
depends	トークンのカンマ区切りリスト		この入力を表示するには、リストに記載されている 1 つまたは複数のトークンが存在している必要があります。1 つまたは複数のトークンを指定することができます。フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
id	テキスト		この入力の ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。
rejects	トークンのカンマ区切りリスト		この入力の表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
searchWhenChanged	論理値		新たに選択された場合にサーチを実行することを示します。
token	文字列		指定された値に置換するサーチ文字列内のトークンを指定します。

親エレメント

<fieldset>

```
<input type="dropdown" token="[search token]"> (1..n)
  <choice> (0..n)
  <label> (0..1)
  <default> (0..1)
  <prefix> (0..1)
  <search> (0..1)
  <selectFirstChoice> (0..1)
  <suffix> (0..1)
```

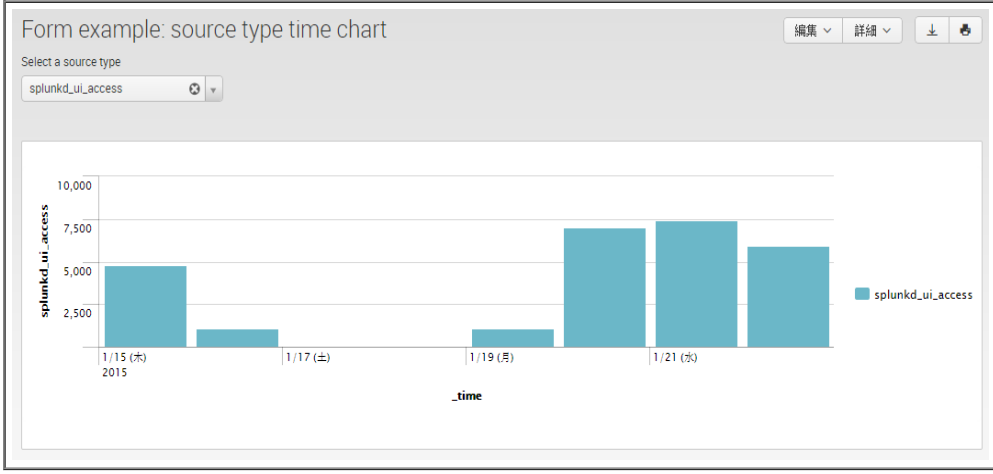
子エレメント

エレメント	タイプ	デフォルト	説明
<allowCustomValues>	論理値		真 (True) の場合、入力のテキスト・フィールドに入力されたカスタム値の選択を有効にします。
<change>	<condition>		条件付きアクションを設定する、入力の選択項目を指定します。 複数選択入力に、<change> エレメントは使用できません。「<change>」を参照してください。
<choice value=[*値*]>	テキスト		**値：必須。**選択肢に使用する値を指定します。 radio または dropdown エレメントの選択項目を指定します。<choice> 指定した値に対して使用するラベル。
<condition>	入力選択項目		条件付きアクションを設定する、入力の選択項目 1 つを指定します。 「<condition> (input)」を参照してください。
<default>	属性値		入力エレメントのデフォルト値を指定します。
<earliest> <latest>	テキスト		もっとも早い/もっとも遅い時間パラメータを示す時間式。入力の選択項目を動的に設定するには、<search> エレメントと一緒に使用します。 時間は相対時間または絶対時間で指定できます。相対時間の場合、相対時間修飾子を使用します。「サーチへの

			時間修飾子の指定」を参照してください。絶対時間の場合、時間は UNIX エポック時フォーマットで指定します。
<fieldForLabel> <fieldForValue>	テキスト		<search> エレメントを使って入力の選択項目を動的に設定する場合に、ラベルと値に対して使用するフィールド。
<label>	テキスト		入力エレメントで表示するテキスト。
<prefix>	テキスト		入力エレメントの値の先頭に付ける文字列。正規表現を使用できます。
<search>	テキスト		入力の選択項目を動的に設定するサーチ。レポートからのサーチを参照するには、<search> エレメントの ref 属性を使用します。「<search>」を参照してください。
<selectFirstChoice>	論理値	false	記載されている最初の項目が、入力のデフォルト項目であることを示します。<default> の値が存在する場合、<selectFirstChoice> は無視されます。
<suffix>	文字列		入力エレメントの値の最後に追加する文字列。正規表現を使用できます。

例

```
<form>
  <label>Form example: source type time chart</label>
  <fieldset autorun="true" submitButton="false">
    <input type="dropdown" token="sourcetype_tok">
      <label>Select a source type</label>
      <default>splunkd</default>
      <choice value="splunkd">splunkd</choice>
      <choice value="splunk_web_access">splunk_web_access</choice>
      <choice value="splunkd_ui_access">splunkd_ui_access</choice>
    </input>
  </fieldset>
  <row>
    <panel>
      <chart>
        <search>
          <query>
            index = _internal sourcetype=$sourcetype_tok$
            | timechart count by sourcetype
          </query>
            <earliest>-7d</earliest>
            <latest>-0d</latest>
        </search>
      </chart>
    </panel>
  </row>
</form>
```



input (複数選択)

<input type="multiselect">

複数選択を受け付けるフォームへの入力を定義します。ユーザーが入力を選択すると、定義された選択肢がドロップダウンリストに表示されます。ユーザーが入力に直接指定して、利用可能な選択肢をフィルタリングすることもできます。

属性

名前	タイプ	デフォルト	説明
depends	トークンのカンマ切り区リスト		この入力を表示するには、リストに記載されている 1 つまたは複数のトークンが存在している必要があります。1 つまたは複数のトークンを指定することができます。フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
id	テキスト		この入力の ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。
rejects	トークンのカンマ切り区リスト		この入力の表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
searchWhenChanged	論理値		新たに選択された場合にサーチを実行することを示します。
token	テキスト		指定された値に置換するサーチ文字列内のトークンを指定します。

親エレメント

<fieldset>

```
<input type="multiselect" token="[search token]"> (1..n)
  <default> (0..1)
  <delimiter> (0..1)
  <label> (0..1)
  <prefix> (0..1)
  <search> (0..1)
  <suffix> (0..1)
  <valuePrefix> (0..1)
  <valueSuffix> (0..1)
```

子エレメント

エレメント	タイプ	デフォルト	説明
<allowCustomValues>	論理値		真 (True) の場合、入力のテキスト・フィールドに入力されたカスタム値の選択を有効にします。
<default>	属性値		入力エレメントのデフォルト値を指定します。
<delimiter>	テキスト	説明を参照	選択した各値の間に配置する文字列です。一般的には、大文字を使って「 OR 」または「 AND 」を指定します。引用符は指定しないでください。また、テキストの前後には、スペース文字を指定してください。 **デフォルト値**：" " デフォルト値に引用符は含まれていません。この引用符は、デフォルト値がスペースであることを示すために、便宜的に用いられています。
<earliest> <latest>	テキスト		もっとも早い/もっとも遅い時間パラメータを示す時間式。入力の選択項目を動的に設定するには、<search> エレメントと一緒に使用します。 時間は相対時間または絶対時間で指定できます。相対時間の場合、相対時間修飾子を使用します。「サーチへの時間修飾子の指定」を参照してください。絶対時間の場

			合、時間は UNIX エポック時フォーマットで指定します。
<fieldForLabel> <fieldForValue>	テキスト		<search> エレメントを使って入力の選択項目を動的に設定する場合に、ラベルと値に対して使用するフィールド。
<label>	テキスト		入力エレメントで表示するテキスト。
<prefix>	テキスト		入力エレメントの値の先頭に付ける文字列。正規表現を使用できます。
<search>	テキスト		入力の選択項目を動的に設定するサーチ。レポートからのサーチを参照するには、<search> エレメントの `ref` 属性を使用します。「<search>」を参照してください。
<suffix>	テキスト		入力エレメントの値の最後に追加する文字列。正規表現を使用できます。
<valuePrefix>	テキスト	"	入力エレメントの値の先頭に付ける文字列。正規表現を使用できます。 デフォルト値は左二重引用符になります (")。
<valueSuffix>	テキスト	"	入力エレメントの値の最後に追加する文字列。正規表現を使用できます。 デフォルト値は右二重引用符になります (")。

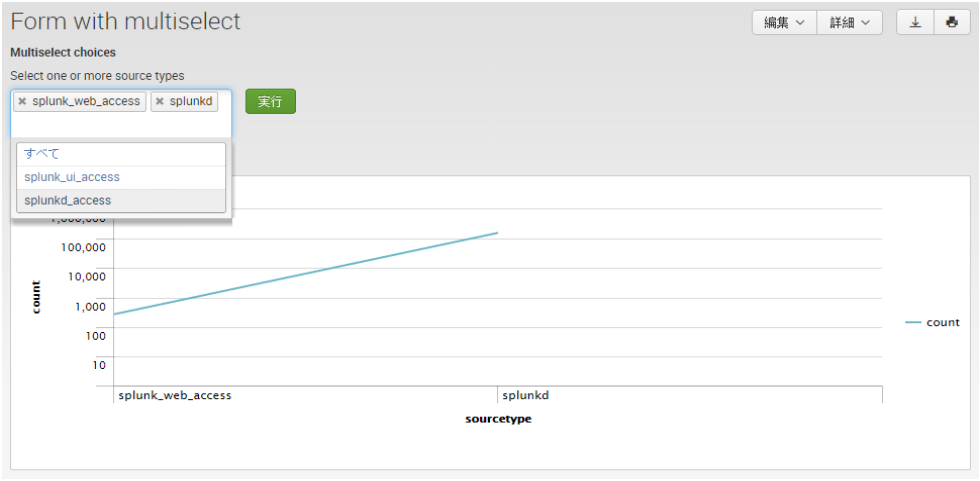
例

この例は、ユーザーが splunkd および splunk_web_access を選択した時に、サーチ用の以下の複数選択文字列を生成します。

(sourcetype ="splunkd" OR sourcetype ="splunk_web_access")

```
<form>
  <label>Form with multiselect</label>
  <fieldset autoRun="false" submitButton="true">
    <html>
      <p>
        <strong>Multiselect choices</strong>
      </p>
    </html>
    <input type="multiselect" token="sourcetype_tok" searchWhenChanged="false">
      <label>Select one or more source types</label>
      <choice value="*">All</choice>
      <choice value="splunk_web_access">splunk_web_access</choice>
      <choice value="splunkd">splunkd</choice>
      <choice value="splunk_ui_access">splunk_ui_access</choice>
      <choice value="splunkd_access">splunkd_access</choice>

      <!--      Build multi-selection search:
        (sourcetype ="value1" OR sourcetype ="value2" OR ...)
      -->
      <prefix>(</prefix>
      <valuePrefix>sourcetype ="</valuePrefix>
      <valueSuffix>"</valueSuffix>
      <delimiter> OR </delimiter>
      <suffix>)</suffix>

    </input>
  </fieldset>
  <row>
    <panel>
      <title></title>
      <chart>
        <search>
          <query>index =_internal $sourcetype_tok$ | stats count by sourcetype</query>
          <earliest>-24h</earliest>
          <latest>now</latest>
        </search>
        <option name="charting.chart">line</option>
        <option name="charting.axisY.scale">log</option>
      </chart>
```

```
    </panel>
  </row>
</form>
```



入力 (ラジオ)

<input type="radio">
フォームのラジオ入力を定義します。

属性

名前	タイプ	デフォルト	説明
depends	トークンのカンマ区切りリスト		この入力を表示するには、リストに記載されている 1 つまたは複数のトークンが存在している必要があります。1 つまたは複数のトークンを指定することができます。フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
id	テキスト		この入力の ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。
rejects	トークンのカンマ区切りリスト		この入力の表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
searchWhenChanged	論理値		新たに選択された場合にサーチを実行することを示します。
token	文字列		指定された値に置換するサーチ文字列内のトークンを指定します。

親エレメント

<fieldset>

```
<input type="radio" token="[search token]"> (1..n)
  <choice> (0..n)
  <label> (0..1)
  <default> (0..1)
  <prefix> (0..1)
  <search> (0..1)
  <selectFirstChoice> (0..1)
```

<suffix> (0..1)

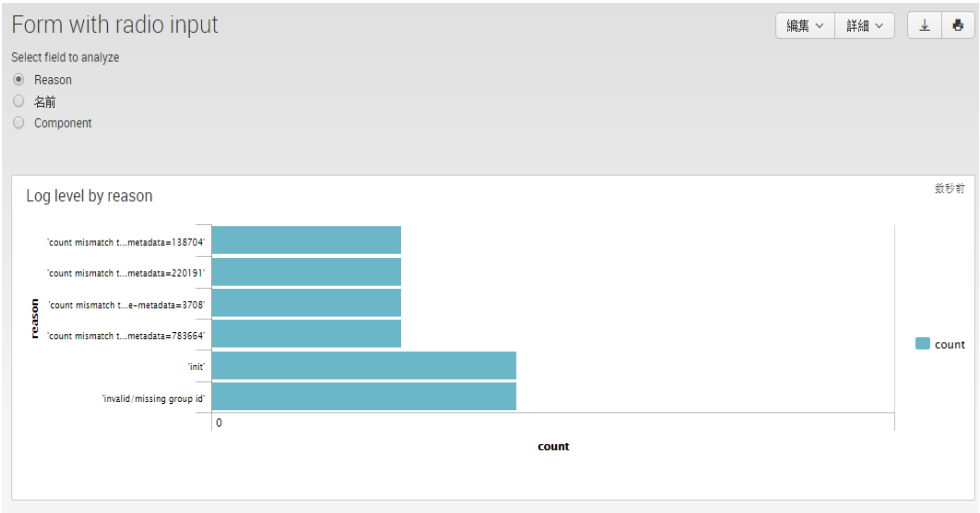
子エレメント

エレメント	タイプ	デフォルト	説明
<change>	<condition>		条件付きアクションを設定する、入力の選択項目を指定します。「<change>」を参照してください。
<choice value=[*値*]>	テキスト		**値：必須。**選択肢に使用する値を指定します。 radio または dropdown エレメントの選択項目を指定します。<choice> 指定した値に対して使用するラベル。
<condition>	入力選択項目		条件付きアクションを設定する、入力の選択項目 1 つを指定します。 「<condition> (input)」を参照してください。
<default>	属性値		入力エレメントのデフォルト値を指定します。
<earliest> <latest>	テキスト		もっとも早い/もっとも遅い時間パラメータを示す時間式。入力の選択項目を動的に設定するには、<search> エレメントと一緒に使用します。 時間は相対時間または絶対時間で指定できます。相対時間の場合、相対時間修飾子を使用します。「サーチへの時間修飾子の指定」を参照してください。絶対時間の場合、時間は UNIX エポック時フォーマットで指定します。
<fieldForLabel> <fieldForValue>	テキスト		<search> エレメントを使って入力の選択項目を動的に設定する場合に、ラベルと値に対して使用するフィールド。
<label>	テキスト		入力エレメントで表示するテキスト。
<prefix>	文字列		入力エレメントの値の先頭に付ける文字列。正規表現を使用できます。
<search>	テキスト		入力の選択項目を動的に設定するサーチ。レポートからのサーチを参照するには、<search> エレメントの `ref` 属性を使用します。「<search>」を参照してください。
<selectFirstChoice>	論理値	false	記載されている最初の項目が、入力のデフォルト項目であることを示します。<default> の値が存在する場合、<selectFirstChoice> は無視されます。
<suffix>	文字列		入力エレメントの値の最後に追加する文字列。正規表現を使用できます。

例

```
<form>
  <label>Form with radio input</label>
  <description></description>
  <fieldset autoRun="True" submitButton="false">
    <input type="radio" token="field_tok">
      <label>Select field to analyze</label>
      <default>component</default>
      <choice value="reason">Reason</choice>
      <choice value="name">Name</choice>
      <choice value="component">Component</choice>
    </input>
  </fieldset>
  <row>
    <panel>
      <chart>
        <title>Log level by $field_tok$</title>
        <search>
          <query>
             index=_internal source=*splunkd.log | stats count by $field_tok$
          </query>
          <earliest>-30d</earliest>
          <latest>now</latest>
        </search>
        <option name="charting.axisY.scale">log</option>
        <option name="charting.chart">bar</option>
      </chart>
```

```
    </panel>
  </row>
</form>
```



入力 (テキスト)

<input type="text">
フォームのテキスト入力を定義します。

属性

名前	タイプ	デフォルト	説明
depends	トークンのカンマ区切りリスト		この入力を表示するには、リストに記載されている 1 つまたは複数のトークンが存在している必要があります。1 つまたは複数のトークンを指定することができます。フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
id	テキスト		この入力の ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。
rejects	トークンのカンマ区切りリスト		この入力の表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
searchWhenChanged	論理値		新しいテキストの入力時にサーチを実行することを指定します。
token	文字列		指定された値に置換するサーチ文字列内のトークンを指定します。

親エレメント

<fieldset>

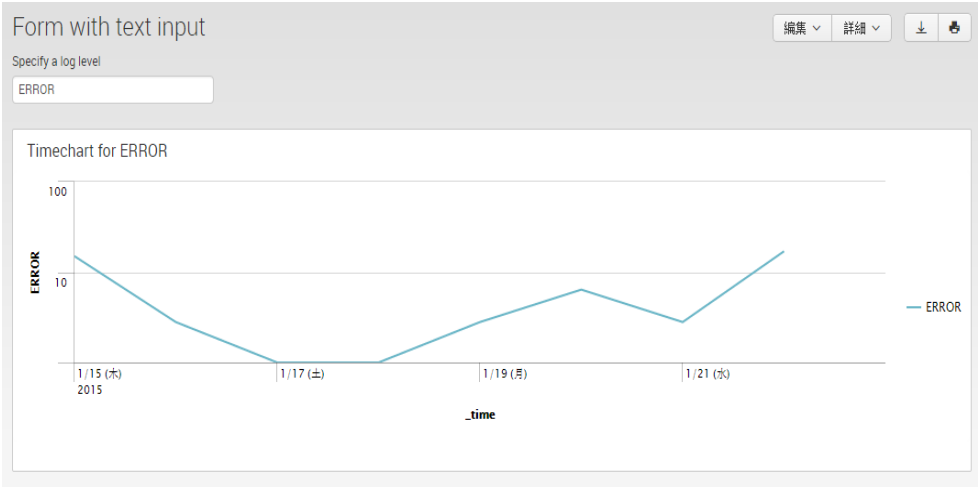
```
<input type="text" token="[search token]"> (1)
  <label> (0..1)
  <default> (0..1)
  <seed> (0..1)
  <prefix> (0..1)
  <suffix> (0..1)
```

子エレメント

エレメント	タイプ	デフォルト	説明
<change>	<condition>		条件付きアクションを設定する、入力の選択項目を指定します。「<change>」を参照してください。
<condition>	入力選択項目		条件付きアクションを設定する、入力の選択項目 1 つを指定します。 「<condition> (input)」を参照してください。
<default>	属性値		入力エレメントのデフォルト値を指定します。
<label>	テキスト		入力エレメントで表示するテキスト。
<prefix>	文字列		入力エレメントの値の先頭に付ける文字列。正規表現を使用できます。
<seed>	属性値		入力エレメントの初期値。
<suffix>	文字列		入力エレメントの値の最後に追加する文字列。正規表現を使用できます。

例

```
<form>
  <label>Form with text input</label>
  <description></description>
  <fieldset autoRun="True" submitButton="false">
    <input type="text" token="log_level_tok">
      <label>Specify a log level</label>
      <default>INFO</default>
    </input>
  </fieldset>
  <row>
    <panel>
      <chart>
        <title>Timechart for $log_level_tok$</title>
        <search>
          <query>
             index=_internal source=*splunkd.log log_level="$log_level_tok$"
             | timechart count by log_level
          </query>
          <earliest>-7d</earliest>
          <latest>now</latest>
        </search>
        <option name="charting.axisY.scale">log</option>
        <option name="charting.chart">line</option>
      </chart>
    </panel>
  </row>
</form>
```



入力 (時間)

<input type="time">
フォームへのタイムレンジピッカー入力を指定します。 複数のタイムレンジピッカーを指定する場合、トークンを使用します。タイムピッカー用のトークンを指定しない場合、タイムピッカーはグローバルになります。タイムピッカートークンの参照またはコード内への直接の指定による時間範囲を指定しない視覚エフェクトは、グローバルタイムピッカーから時間範囲を適用します。

属性

名前	タイプ	デフォルト	説明
depends	トークンのカンマ区切りリスト		この入力を表示するには、リストに記載されている 1 つまたは複数のトークンが存在している必要があります。1 つまたは複数のトークンを指定することができます。フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
id	テキスト		この入力の ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。
rejects	トークンのカンマ区切りリスト		この入力の表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
token	テキスト		タイムレンジピッカーをパネルに関連付けるには、トークンを使用します。 タイムピッカートークンを参照する場合、トークンに earliest および latest 修飾子を使用して、時間範囲を指定します。以下の例を参照してください。
searchWhenChanged	論理値		新たに選択された場合にサーチを実行することを示します。

親エレメント

<fieldset>

```
<input type="time" [ token="[text]" ] [ searchWhenChanged="[true|false]" ]> (0..n)
  <label> (0..1)
  <default> (0..1)
    [time preset] (0..1) |
    <earliest> (0..1)
    <latest> (0..1)
  </default>
```

子エレメント

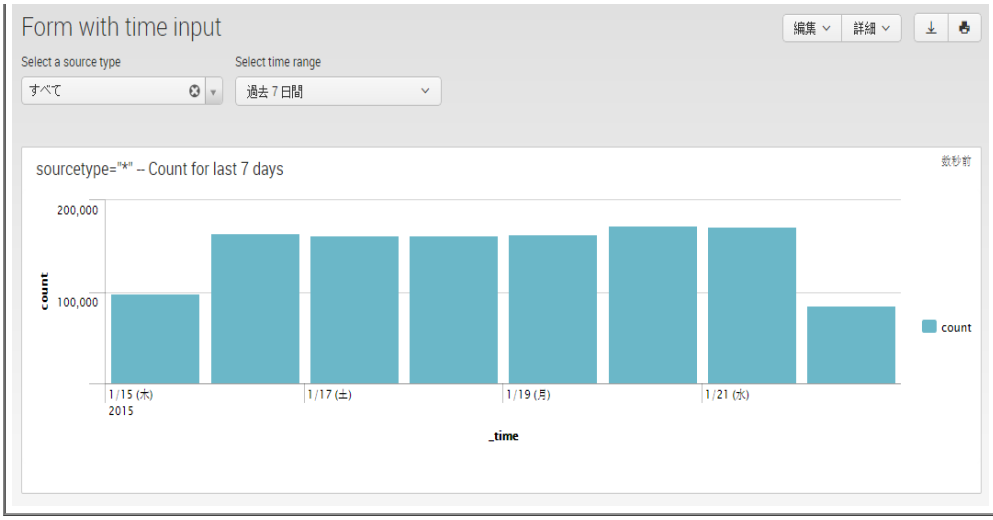
エレメント	タイプ	デフォルト	説明
<change>	<condition>		条件付きアクションを設定する、入力の選択項目を指定します。 複数選択入力に、<change> エレメントは使用できません。「<change>」を参照してください。
<condition>	入力選択項目		条件付きアクションを設定する、入力の選択項目 1 つを指定します。 「<condition> (input)」を参照してください。
			もっとも早い/もっとも遅い時間パラメータを示す時間

		式。
<earliest> <latest>	テキスト	時間は相対時間または絶対時間で指定できます。相対時間の場合、相対時間修飾子を使用します。「サーチへの時間修飾子の指定」を参照してください。絶対時間の場合、時間は UNIX エポック時フォーマットで指定します。
<default>	テキスト または 時間修飾子	入力エレメントのデフォルト値を指定します。 times.conf に設定されているプリセット値を使用できます。 または 独自のデフォルト時間範囲の <earliestTime> (もっとも早い時間) と <latestTime> (もっとも遅い時間) を使用できます。 詳細は「<earliestTime>」および「<latestTime>」を参照してください。
<label>	テキスト	入力エレメントで表示するテキスト。

例

タイムピッカーのデフォルト値は、過去 7 日間に設定されています。この例の <chart> エレメントは、タイムピッカーの $time_tok$ トークンを参照しています。新たに時間範囲が選択されると、グラフが更新されます。

```
<form>
  <label>Form with time input</label>
  <description/>
  <fieldset submitButton="false">
    <input type="dropdown" token="source_tok" searchWhenChanged="true">
      <label>Select a source type</label>
      <choice value="*">All</choice>
      <search>
        <query>
          index=_internal | stats count by sourcetype
        </query>
        <earliest>-7d@h</earliest>
        <latest>now</latest>
      </search>
      <fieldForLabel>sourcetype</fieldForLabel>
      <fieldForValue>sourcetype</fieldForValue>
      <prefix>sourcetype="</prefix>
      <suffix>"</suffix>
      <default>*</default>
    </input>
    <input type="time" token="time_tok" searchWhenChanged="true">
      <label>Select time range</label>
      <default>
        <earliest>-7d@h</earliest>
        <latest>now</latest>
      </default>
    </input>
  </fieldset>
  <row>
    <panel>
      <chart>
        <title>$source_tok$ -- Count for last 7 days</title>
        <search>
          <query>
            index=_internal $source_tok$ | timechart count
          </query>
          <earliest>$time_tok.earliest$</earliest>
          <latest>$time_tok.latest$</latest>
        </search>
        <option name="charting.chart">column</option>
      </chart>
    </panel>
  </row>
</form>
```



change

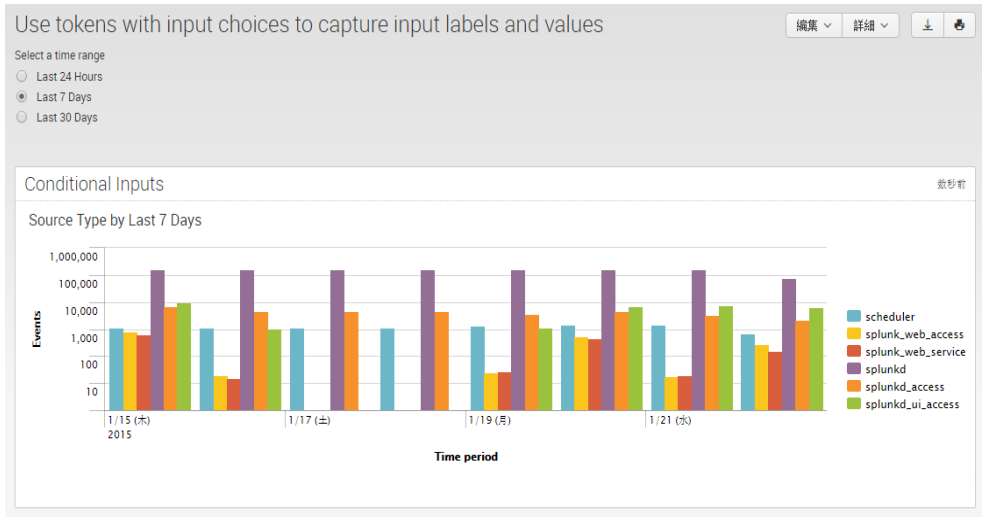
<change>
フォーム入力の選択された項目に基づいて、トークンを設定します。<condition> エレメントと一緒に使用して、選択項目に基づく条件付きアクションを定義できます。 複数選択入力に、<change> エレメントは使用できません。
親エレメント <input type="checkbox"> <input type="dropdown"> <input type="radio"> <input type="text"> <input type="time">

`<change>`
`  <condition>(0..n)`
`  (<link> | <set> | <unset>) (1..n)`

属性 このエレメントの属性はありません。

例
 	
入力から選択されたラベルと値を取得するには、<change> エレメントを使用します。
 	
`<form>`
`  <label>Use tokens with input choices to capture input labels and values</label>`
`  <fieldset submitButton="false">`
`  <input type="radio" token="period_tok">`
`  <label>Select a time range</label>`
`  <choice value="-24h@h">Last 24 Hours</choice>`
`  <choice value="-7d@d">Last 7 Days</choice>`
`  <choice value="-30d@d">Last 30 Days</choice>`
`  <default>Last 24 Hours</default>`
 	
`  <change>`
`  <!-- use predefined input tokens to set -->`
`  <!-- tokens for the selected label and value -->`
`  <set token="date_label">$label$</set>`
`  <set token="earliest_tok">$value$</set>`
`  </change>`
 	
`  </input>`
`  </fieldset>`

```
  <row>
    <panel>
      <title>Conditional Inputs</title>
      <chart>
        <!-- Display selected label in the title -->
        <title>Source Type by $date_label$</title>

        <search>
          <query>index = _internal | timechart count by sourcetype</query>
          <!-- use the value of earliest_tok -->
          <!-- to set the time range         -->
          <earliest>$earliest_tok$</earliest>
          <latest>now</latest>
        </search>

        <option name="charting.axisY.scale">log</option>
        <option name="charting.axisTitleX.text">Time period</option>
        <option name="charting.axisTitleY.text">Events</option>
      </chart>
    </panel>
  </row>
</form>
```



condition (input)

<condition>
入力の選択項目に基づいて、アクションの範囲を指定します。親エレメントの <change> が存在しない場合は、すべての選択項目にアクションが適用されます。複数選択入力に、<condition> エレメントは使用できません。 **注意：**<condition> エレメントは、input エレメントと drilldown エレメントの両方に適用されます。詳細は「<condition> (drilldown)」を参照してください。
親エレメント <change>
`<condition>` `  (<link> | <set> | <unset>) (1..n)`

属性

名前	タイプ	デフォルト	説明
label	テキスト	*	条件を適用する入力 <label> エレメントを指定します。 「*」の場合、すべての入力 <label> エレメントに条件を適用します。

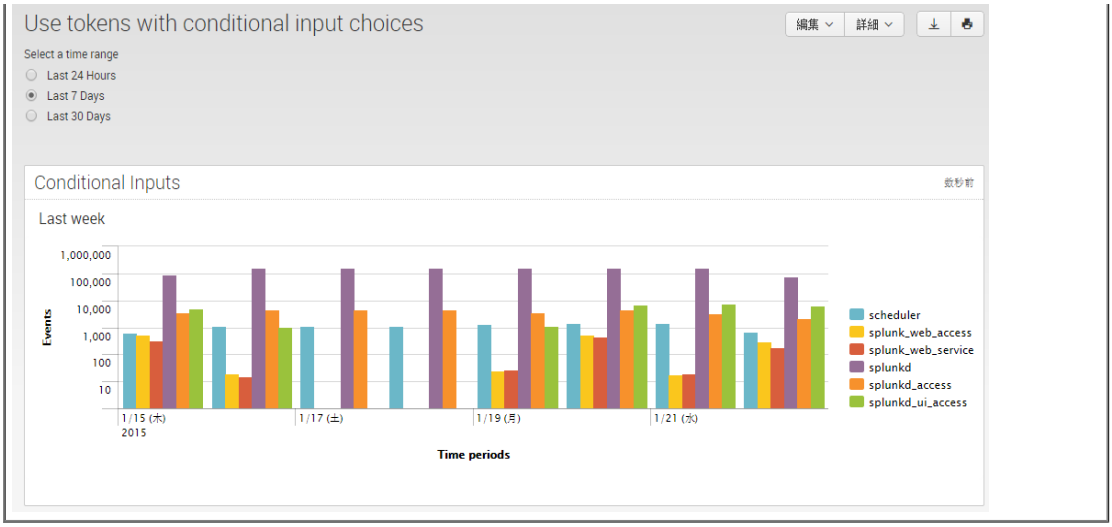
値	テキスト	*	条件を適用する入力 <value> エレメントを指定します。 「*」の場合、すべての入力 <value> エレメントに条件を適用します。
field	テキスト	*	**ドリルダウン・コンテキストのみ。**ドリルダウンを実装する、またはトークンを設定/解除するサーチフィールドを指定します。「<condition> (drilldown)」を参照してください。

例

サーチのプリセット時間範囲を選択するには、条件付き入力を使用します。

選択項目のトークンが、グラフのタイトルに表示されます。選択された値の条件付きトークンが、グラフのデータを提供します。

```
<form>
  <label>Use tokens with conditional input choices</label>
  <fieldset submitButton="false">
    <input type="radio" token="period_tok">
      <label>Select a time range</label>
      <choice value="-24h@h">Last 24 Hours</choice>
      <choice value="-7d@h">Last 7 Days</choice>
      <choice value="-30d@h">Last 30 Days</choice>
      <default>Last 24 Hours</default>

      <!-- set condition based on the label defined by <choice> -->
      <!-- Within each condition, specify a custom label for display -->
      <!-- Capture the selected value in the token, earliest_tok -->
      <change>
        <condition label="Last 24 Hours">
          <set token="date_label">Yesterday</set>
          <set token="earliest_tok">$value$</set>
        </condition>
        <condition label="Last 7 Days">
          <set token="date_label">Last week</set>
          <set token="earliest_tok">$value$</set>
        </condition>
        <condition label="Last 30 Days">
          <set token="date_label">Last month</set>
          <set token="earliest_tok">$value$</set>
        </condition>
      </change>
    </input>
  </fieldset>
  <row>
    <panel>
      <title>Conditional Inputs</title>
      <chart>

        <!-- Display selected label in the title -->
        <title>$date_label$</title>

        <search>
          <query>index = _internal | timechart count by sourcetype</query>
          <!-- use the value of earliest_tok -->
          <!-- to set the time range         -->
          <earliest>$earliest_tok$</earliest>
          <latest>now</latest>
        </search>

        <option name="charting.axisY.scale">log</option>
        <option name="charting.axisTitleX.text">Time periods</option>
        <option name="charting.axisTitleY.text">Events</option>
      </chart>
    </panel>
  </row>
</form>
```



パネルの視覚化エレメント

chart

<chart>

サーチデータをグラフ形式で表示するパネル。パネルに使用するサーチには、インラインサーチまたはグラフ書式パラメータを含む保存済みレポートを利用できます。レポートの保存の詳細は、「レポートの作成と編集」を参照してください。

グラフパネルに保存済みレポートをロードする場合、保存済みレポートの書式設定もロードされます。ただし、グラフの書式設定はグラフオプションで上書きすることができます。

グラフは名前付きオプションを使って、グラフ固有のプロパティを指定します。この参照は、グラフ用の基本的なパネルオプションを記載しています。グラフオプションの全リストについては、「グラフ設定リファレンス」を参照してください。

属性

名前	タイプ	デフォルト	説明
depends	トークンのカンマ区切りリスト		この視覚エフェクトを表示するには、リストに記載されている 1 つまたは複数のトークンが存在している必要があります。1 つまたは複数のトークンを指定することができます。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
id	テキスト		視覚エフェクトの ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。 以下の用語は内部使用向けに予約されており、`id` に使用することはできません。 dashboard, search, default, submitted, footer, url, header
rejects	トークンのカンマ区切りリスト		この視覚エフェクトの表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。

親エレメント

<row>
<panel>

```
<chart>
  <title> (0..1)
  <search> (0..1)
    <earliest> (0..1)
    <latest> (0..1)
  <drilldown> (0..n)
  <selection> (0..n, for charts of type area, line, and column only)
  <option name="[property]"> (0..n)
```

オプション

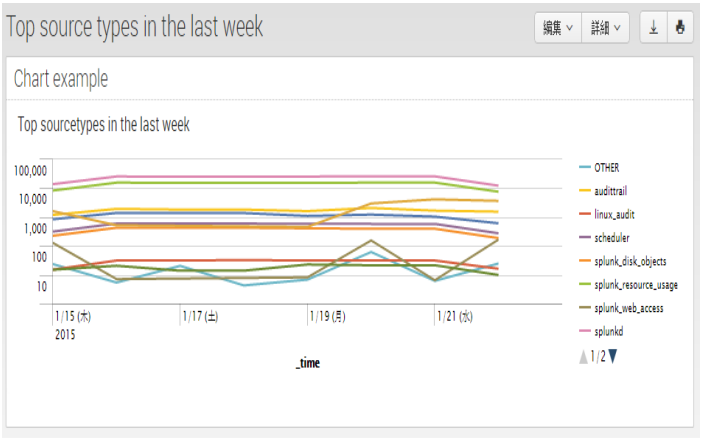
プロパティ	タイプ	デフォルト	説明
charting.chart	(area | bar | column | fillerGauge | line | markerGauge | pie | radialGauge | scatter)	列	グラフタイプを設定します。
charting.legend.placement	(top | left | bottom | right | none)	右	凡例の配置を示します。
charting.*	—	—	グラフがサポートするすべての書式設定オプション。詳細は、「カスタムグラフ設定リファレンス」を参照してください。
drilldown	(all | none)	—	**非推奨。**『グラフ設定リファレンス』の「全般的なグラフのプロパティ」に記載されている、charting.drilldown を使用します。
height	数値	—	グラフの高さ (ピクセル)。
link.exportResults.visible	論理値	(説明を参照)	パネルの下部に [エクスポート] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.inspectSearch.visible	論理値	(説明を参照)	パネルの下部に [調査] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.openPivot.visible	論理値	(説明を参照)	パネルの下部に [ピボットで開く] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.openSearch.search	サーチ文字列	—	[サーチで開く] ボタンで使用する代替サーチ。
link.openSearch.searchEarliestTime	(時間修飾子)	(説明を参照)	link.openSearch.search で指定する代替サーチで使用する、もっとも早い時間。 **デフォルト値**：パネルが使用するもっとも早い時間。 時間修飾子を使って時間を指定します。サーチの時間修飾子については、「時間修飾子の指定」を参照してください。
link.openSearch.searchLatestTime	(時間修飾子)	(説明を参照)	link.openSearch.search で指定する代替サーチで使用する、もっとも遅い時間。 **デフォルト値**：パネルが使用するもっとも遅い時間。 時間修飾子を使って時間を指定します。サーチの時間修飾子については、「時間修飾子の指定」を参照してください。

link.openSearch.text	テキスト	サーチで開く	[サーチで開く] ボタンで使用するラベル。
link.openSearch.ViewTarget	ビュー名	サーチ	[サーチで開く] ボタンのターゲットビュー。
link.openSearch.visible	論理値	(説明を参照)	パネルの下部に [サーチで開く] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.visible	論理値	true	パネルの下部にリンクボタンを表示します。
refresh.auto.interval	数値	0	更新間隔を秒で指定します。パネルの更新を無効にする場合は、0 (または負の整数) を指定します。
refresh.time.visible	論理値	true	パネルに更新時間インジケータを表示します。
refresh.link.visible	論理値	true	パネルに更新リンクを表示します。

例

インラインサーチを使った折れ線グラフパネルの例。これは、結果を指定した時間ウィンドウに制限し、X 軸と Y 軸のラベルを提供しています。

```
<dashboard>
  <label>Top source types in the last week</label>
  <row>
    <panel>
      <title>Chart example</title>
      <chart>
        <title>Top sourcetypes in the last week</title>
        <search>
          <query>
          index=_internal source="*metrics.log" group=per_sourcetype_thruput
          | timechart sum(kb) by series
          </query>
          <earliest>-1w</earliest>
          <latest>now</latest>
        </search>
        <option name="height">200px</option>
        <option name="charting.chart">line</option>
        <option name="charting.axisY.scale">log</option>
        <option name="charting.chart.nullValueMode">connect</option>
       </chart>
    </panel>
    . . .
  </row>
</dashboard>
```



イベント

<イベント>
サーチ結果を個別のイベントとして表示するパネル。

属性

名前	タイプ	デフォルト	説明
depends	トークンのカンマ区切りリスト		この視覚エフェクトを表示するには、リストに記載されている 1 つまたは複数のトークンが存在している必要があります。1 つまたは複数のトークンを指定することができます。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
id	テキスト		視覚エフェクトの ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。 以下の用語は内部使用向けに予約されており、`id` に使用することはできません。 dashboard, default, footer, header, search, submitted, url
rejects	トークンのカンマ区切りリスト		この視覚エフェクトの表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。

親エレメント

<row>
<panel>

```
<event>
  <title> (0..1)
  <search> (0..1)
    <earliest> (0..1)
    <latest> (0..1)
  <fields> (0..1)
  <option name="[property]"> (0..n)
```

オプション

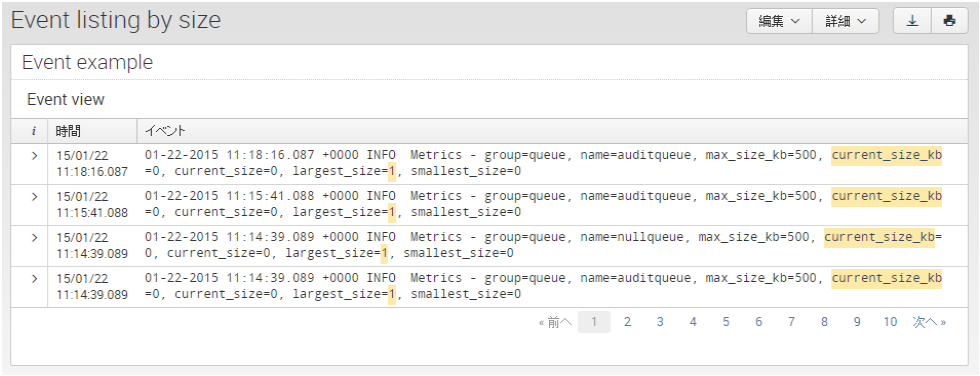
プロパティ	タイプ	デフォルト	説明
count	整数		表示する最大行数。
displayRowNumbers	論理値	False	(廃止予定) 属性 rowNumbers を使用します。 行番号の表示を切り替えます。
drilldown	(all | none)	all	すべてのタイプ固有のドリルダウン (list.drilldown、table.drilldown、raw.drilldown) を有効 (または) 無効にします。 タイプ固有のドリルダウンオプションは、ここの設定に優先します。 all：ドリルダウン有効。 none：ドリルダウン無効。
			イベントを表示するか、または結果を表示するかを切り替えます。

entityName	(events | results)	イベント	イベントは個別のイベントですが、結果は統計演算子により作成されます。
link.exportResults.visible	論理値	(説明を参照)	パネルの下部に [エクスポート] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.inspectSearch.visible	論理値	(説明を参照)	パネルの下部に [調査] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.openPivot.visible	論理値	(説明を参照)	パネルの下部に [ピボットで開く] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.openSearch.search	サーチ文字列	—	[サーチで開く] ボタンで使用する代替サーチ。
link.openSearch.searchEarliestTime	(時間修飾子)	(説明を参照)	link.openSearch.search で指定する代替サーチで使用する、もっとも早い時間。 **デフォルト値**：パネルが使用するもっとも早い時間。 時間修飾子を使って時間を指定します。 サーチの時間修飾子については、「時間修飾子の指定」を参照してください。
link.openSearch.searchLatestTime	(時間修飾子)	(説明を参照)	link.openSearch.search で指定する代替サーチで使用する、もっとも遅い時間。 **デフォルト値**：パネルが使用するもっとも遅い時間。 時間修飾子を使って時間を指定します。 サーチの時間修飾子については、「時間修飾子の指定」を参照してください。
link.openSearch.text	テキスト	サーチで開く	[サーチで開く] ボタンで使用するラベル。
link.openSearch.ViewTarget	ビュー名	サーチ	[サーチで開く] ボタンのターゲットビュー。
link.openSearch.visible	論理値	(説明を参照)	パネルの下部に [サーチで開く] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.visible	論理値	true	パネルの下部にリンクボタンを表示します。
list.drilldown	(full | inner | outer | none)	完全	イベントリスト内のドリルダウン操作を指定します。 full：エントリ全体のドリルダウンを有効にします。 inner：イベントリストの内部エレメントのドリルダウンを有効にします。 outer：イベントリストの外部エレメントのドリルダウンを有効にします。 none：ドリルダウンを無効にします。
list.wrap	論理値	true	イベントリストのコンテンツを折り返し表示するかどうかを示します。
maxLines	整数		各結果/イベントに対して表示する最大行数です。
			raw イベントリスト内のドリルダウン操作を指定します。 full：エントリ全体のドリルダウンを有効に

			します。
raw.drilldown	(full | inner | outer | none)	完全	**inner**：イベントリストの内部エレメントのドリルダウンを有効にします。 **outer**：イベントリストの外部エレメントのドリルダウンを有効にします。 **none**：ドリルダウンを無効にします。
refresh.auto.interval	数値	0	更新間隔を秒で指定します。 パネルの更新を無効にする場合は、0 (または負の整数) を指定します。
refresh.time.visible	論理値	true	パネルに更新時間インジケータを表示します。
refresh.link.visible	論理値	true	パネルに更新リンクを表示します。
rowNumbers	論理値	False	行番号を表示するかどうかを示します。
segmentation	(none | inner | outer | full)	なし	**廃止予定**：代わりに **list.drilldown** または **raw.drilldown** を使用してください。 表示するイベントのセグメントを設定します。 これは、イベント内でクリックできる項目に影響します。 list.drilldown または raw.drilldown と一緒にセグメントを指定すると、segmentation の値は無視されます。
showPager	論理値	True	ページャーをオン/オフにします。
softWrap	論理値		イベントの折り返し表示を有効にします。
table.sortColumn	テキスト		テーブルのソートする列を指定します。
table.sortDirection	(asc | desc)	昇順	テーブル内の項目のソート方向を示します。
table.drilldown	(all | none)	all	テーブルのドリルダウン機能が有効かどうかを示します。 **all**：ドリルダウン有効。 **none**：ドリルダウン無効。
table.wrap	論理値		テーブル内のテキストを折り返し表示するかどうかを示します。
type	(list | raw | table)	list	イベントの表示形式を示します。

例

```
<dashboard>
  <label>Event listing by size</label>
  <row>
    <panel>
      <title>Event example</title>
      <event>
        <title>Event view</title>
        <search>
          <query>
          index = _internal current_size_kb < 1
          </query>
          <earliest>-1w</earliest>
          <latest>now</latest>
        </search>
        <option name="showPager">true</option>
        <option name="count">4</option>
        <option name="rowNumbers">false</option>
      </event>
    </panel>
  </row>
</dashboard>
```



html

<html>

HTML パネルには、インライン HTML が表示されます。このパネルは、HTML タグ内のコンテンツすべてを文字通りに解釈して、パネルに HTML で書式設定されたテキストを表示します。

画像などの相対リンク山椒は、現在のビューの場所からの相対的な位置となります。HTML パネルで利用できるオプションはありません。

属性

名前	タイプ	デフォルト	説明
depends	トークンのカンマ区切りリスト		この視覚エフェクトを表示するには、リストに記載されている 1 つまたは複数のトークンが存在している必要があります。1 つまたは複数のトークンを指定することができます。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
encoded	論理値	false	内部使用専用。真 (True) の場合、ダッシュボードは XML コンテンツの代わりに、デコードされたコンテンツを使用します。
id	テキスト		視覚エフェクトの ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。 以下の用語は内部使用向けに予約されており、`id` に使用することはできません。 dashboard, search, default, submitted, footer, url, header
rejects	トークンのカンマ区切りリスト		この視覚エフェクトの表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
src	文字列		HTML パネルに表示する HTML ファイルを指定します。 HTML ファイルは以下のディレクトリに保管します。 `$SPLUNK_HOME/etc/apps/appname/appserver/static/`
tokens	論理値	true	偽 (False) の場合、<html> パネルのトークン置換が無効になります。

親エレメント

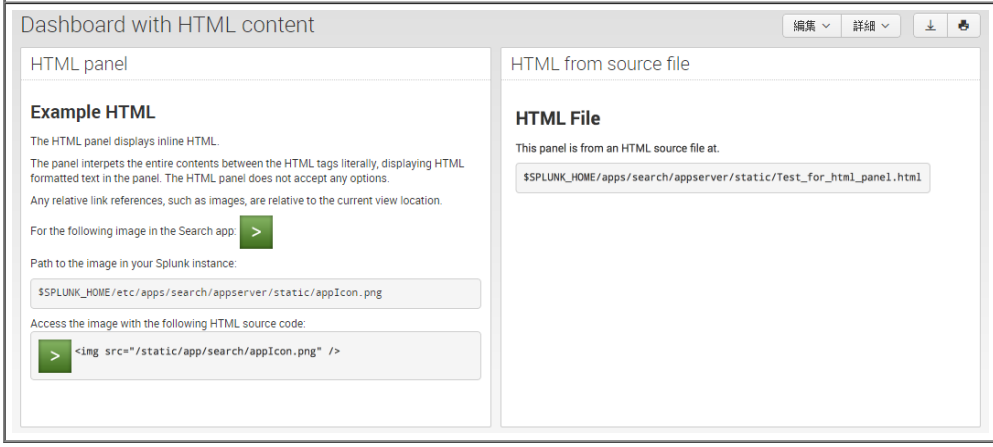
```
<row>
  <panel>
```

```
<html>
```

例

ローカル画像の参照方法を表す HTML パネル。

```
<dashboard>
  <label>Dashboard with HTML content</label>
  <row>
    <panel>
      <title>HTML panel</title>
      <html>
        <h1>Example HTML</h1>
        <p>The HTML panel displays inline HTML.</p>
        <p>
          The panel interpets the entire contents between the HTML tags literally, displaying
          HTML formatted text in the panel. The HTML panel does not accept any options.
        </p>
        <p>
          Any relative link references, such as images,
          are relative to the current view location.
        </p>
        <p>
          For the following image in the Search app: <img src="/static/app/search/appIcon.png"/>
        </p>
        <p>Path to the image in your Splunk instance:
          <pre>$SPLUNK_HOME/etc/apps/search/appserver/static/appIcon.png</pre>
          Access the image with the following HTML source code:
          <pre><img src="/static/app/search/appIcon.png" /></pre>
        </p>
      </html>
    </panel>
    <panel>
      <title>HTML from source file</title>
      <html src="Test_for_html_panel.html" />
    </panel>
  </row>
</dashboard>
```



map

<map>

地理的座標を世界地図上の対話型マーカーとしてのマッピングを提供しています。この視覚エフェクトは、geostats サーチコマンドの結果に依存しています。

geostats サーチの実装方法の詳細は、『サーチリファレンス』の「geostats」を参照してください。

属性

名前	タイプ	デフォルト	説明

depends	トークンのカンマ区切りリスト	この視覚エフェクトを表示するには、リストに記載されている 1 つまたは複数のトークンが存在している必要があります。1 つまたは複数のトークンを指定することができます。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
id	テキスト	視覚エフェクトの ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。 以下の用語は内部使用向けに予約されており、`id` に使用することはできません。 dashboard, search, default, submitted, footer, url, header
rejects	トークンのカンマ区切りリスト	この視覚エフェクトの表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。

親エレメント

<row>
<panel>

```
<map>
  <title> (0..1)
  <search> (0..1)
    <earliest> (0..1)
    <latest> (0..1)
  <option name="[property]"> (0..n)
```

オプション

プロパティ	タイプ	デフォルト	説明
drilldown	(all | none)	all	**all**：ドリルダウン有効。 **none**：ドリルダウン無効。
link.exportResults.visible	論理値	(説明を参照)	パネルの下部に [エクスポート] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.inspectSearch.visible	論理値	(説明を参照)	パネルの下部に [調査] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.openSearch.search	サーチ文字列	—	[サーチで開く] ボタンで使用する代替サーチ。
link.openSearch.searchEarliestTime	(時間修飾子)	(説明を参照)	link.openSearch.search で指定する代替サーチで使用する、もっとも早い時間。 **デフォルト値**：パネルが使用するもっとも早い時間。 時間修飾子を使って時間を指定します。サーチの時間修飾子については、「時間修飾子の指定」を参照してください。
link.openSearch.searchLatestTime	(時間修飾子)	(説明を参照)	link.openSearch.search で指定する代替サーチで使用する、もっとも遅い時間。 **デフォルト値**：パネルが使用するもっとも遅い時間。 時間修飾子を使って時間を指定します。サーチの時間修飾子については、「時間修飾子の指定」を

			参照してください。
link.openSearch.text	テキスト	サーチで開く	[サーチで開く] ボタンで使用するラベル。
link.openSearch.ViewTarget	ビュー名	サーチ	[サーチで開く] ボタンのターゲットビュー。
link.openSearch.visible	論理値	(説明を参照)	パネルの下部に [サーチで開く] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.visible	論理値	true	パネルの下部にリンクボタンを表示します。
mapping.data.maxClusters	整数	100	表示する最大クラスタ数。 **警告：**このオプションに大量のクラスタを設定すると、パフォーマンスが大幅に低下する可能性があります。1000 未満の値を指定することをお勧めします。
mapping.fieldColors	*フィールド:16 進値,* ...		特定のシリーズの色を定義する、フィールド名から 16 進色値 (0xRRGGBB) へのカンマ区切りマップ。
mapping.seriesColors	*16 進値,* . . .	デフォルト*	16 進色値 (0xRRGGBB) のリスト。fieldColors プロパティを使って特定の色が割り当てられていないシリーズ用の、サンプル色を提供しています。
mapping.map.center	(緯度、経度)		地図の初期中心点。緯度値の範囲は -85～85 でこの範囲外の値は省略されます。経度値の範囲は -180～180 で、この範囲外の値はその範囲内にまるように修正されます。
mapping.map.zoom	数値		地図の初期ズームレベル。
mapping.map.fitBounds	(south-lat, west-long, north-lat, east-long)		地図の表示領域内に合わせた初期境界。緯度値の範囲は -85～85 で、この範囲外の値は省略されます。 経度値の範囲は -180～180 で、この範囲外の値はその範囲内に収まるように修正されます。 このプロパティに割り当てられた値は、center たは zoom プロパティに割り当てられた値に優先します。 サンフランシスコ湾岸地域の指定例： <option name="mapping.map.fitBounds">(37.5,-123,38,-122)</option>
mapping.tileLayer.url	URL テンプレート	説明を参照	タイルのリクエストに使用する URL、以下のテンプレートをベースにしています。 http://(s).tile.openstreetmap.org/(z)/(x)/(y).p
mapping.tileLayer.subdomains	[文字列,. . .]	[a,b,c]	タイルリクエストを配布するサブドメインのリスト。サブドメインを多くすると、より多くのタイルを同時にリクエストできます。 以下の例を参照してください。
mapping.tileLayer.minZoom	整数	0	タイルセットの最小ズームレベル。
mapping.tileLayer.maxZoom	整数	7	タイルセットの最大ズームレベル。 最大ズームレベルは、負以外の値で指定します。
mapping.tileLayer.invertY	論理値	False	タイルリクエストの Y 軸を反転するかどうかをします。TMS サーバーは、反転 Y 軸を使用します。
			値の右下に表示する著作権帰属情報。デフォル

			値： Map data (c) 2012 OpenStreetMap contributors, CC-BY-SA. 以下の例を参照してください。
mapping.markerLayer.markerOpacity	数値	0.8	マーカーの不透明度。値は 0 (透明)〜1 (不透明の範囲で指定します。
mapping.markerLayer.markerMinSize	数値	10	マーカーの最小サイズ (ピクセル)。
mapping.markerLayer.markerMaxSize	数値	50	マーカーの最大サイズ (ピクセル)。
refresh.auto.interval	数値	0	更新間隔を秒で指定します。パネルの更新を無効にする場合は、0 (または負の整数) を指定します。
refresh.time.visible	論理値	true	パネルに更新時間インジケータを表示します。
refresh.link.visible	論理値	true	パネルに更新リンクを表示します。

* mapping.seriesColors のデフォルト値：
[0x6CB8CA,0xFAC61D,0xD85E3D,0x956E96,0xF7912C,0x9AC23C,0x5479AF,0x999755,0xDD87B0,0x65AA82,
0xA7D4DF,0xFCDD77,0xE89E8B,0xBFA8C0,0xFABD80,0xC2DA8A,0x98AFCF,0xC2C199,0xEBB7D0,0xA3CCB4,
0x416E79,0x967711,0x823825,0x59425A,0x94571A,0x5C7424,0x324969,0x5C5B33,0x85516A,0x3D664E]

mapping.data.maxClusters example

以下の例は、クラスタの最大数を 250 に設定します。

```
<map>
   <option name="mapping.data.maxClusters">250</option>
</map>
```

mapping.fieldColors および mapping.seriesColors の例

以下の例では、「foo」フィールドと「bar」フィールドをそれぞれ赤 (0xFF0000) と緑 (0x00FF00) に、その他のフィールドを青 (0x0000FF) に設定します。

```
<map>
   <option name="mapping.fieldColors">{foo:0xFF0000,bar:0x00FF00}</option>
   <option name="mapping.seriesColors">[0x0000FF]</option>
</map>
```

mapping.map.fitBounds example

以下の例は、地図ビューをサンフランシスコ周辺地域に初期化します。

```
<map>
  <option name="mapping.map.fitBounds">
    (37.5,-123,38,-122)
  </option>
</map>
```

mapping.tileLayer.* の例

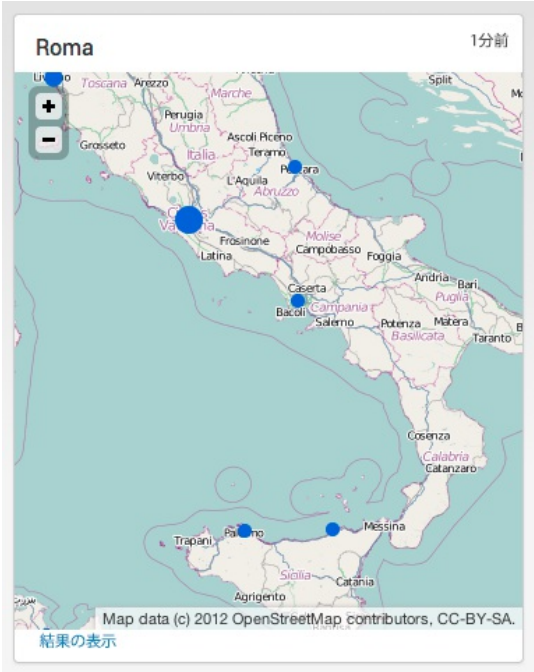
以下の例は、クライアントが openstreetmap.org (デフォルトの設定) からのタイルをリクエストするように設定しています。

```
<map>
   <option name="mapping.tileLayer.url">http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png</option>
   <option name="mapping.tileLayer.subdomains">[a,b,c]</option>
   <option name="mapping.tileLayer.maxZoom">18</option>
   <option name="mapping.tileLayer.attribution">
     Map data (c) 2012 OpenStreetMap contributors, CC-BY-SA.
   </option>
</map>
```

map の例、foursquare データを使用

この例は、foursquare データをソース `foursquare` としてインデックスを作成することを前提にしています。これは、以下に記載されている地図を生成します。

```
<map>
  <title>Roma</title>
  <searchString>
    sourcetype=foursquare
    | geostats latfield=checkin.geolat longfield=checkin.geolong count by checkin.user.gender
  </searchString>
  <option name="mapping.data.maxClusters">500</option>
  <option name="mapping.markerLayer.markerMaxSize">20</option>
  <option name="mapping.map.fitBounds">(41.3,12.7,41.5,12.8)</option>
  <option name="mapping.seriesColors">[0x0060DD]</option>
  <option name="mapping.map.zoom">4</option>
</map>
```



単数形

<単数形>

単一値を返すサーチの結果を表示するパネル。返された値の rangemap を指定することで、結果の色を変更することができます。

警告：複数値を返すサーチを指定した場合、単一値パネルには返されたデータの最初の行または列の値が表示されます。

属性

名前	タイプ	デフォルト	説明
depends	トークンのカンマ区切りリスト		この視覚エフェクトを表示するには、リストに記載されている 1 つまたは複数のトークンが存在している必要があります。1 つまたは複数のトークンを指定することができます。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
id	テキスト		視覚エフェクトの ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。 以下の用語は内部使用向けに予約されており、`id` に使用することはできません。 dashboard, search, default, submitted, footer, url

		header
rejects	トークンのカンマ区切りリスト	この視覚エフェクトの表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。

親エレメント

<row>
<panel>

```
<single>
  <title> (0..1)
  <search> (0..1)
    <earliest> (0..1)
    <latest> (0..1)
  <option name="[property]"> (0..n)
```

オプション

プロパティ	タイプ	デフォルト	説明
additionalClass	CSS クラス名		結果コンテナに追加する他の css クラス名。
afterLabel	文字列		結果の後に表示するラベル。
beforeLabel	文字列		結果の前に表示するラベル。
classField	(*classname* | severe | high | elevated | guarded | low | None)		最初の結果の classField の値を、追加の CSS クラスとして結果コンテナに追加します。 CSS クラス名を指定するか、またはあらかじめ定義されているクラス (severe、high、elevated、guarded、low、None) を使用します。
drilldown	(all | none)	なし	**all**：ドリルダウン有効。 **none**：ドリルダウン無効。 このオプションは、単一値の動的ドリルダウンを導入するために、<drilldown> エレメントに適用されます。 <single> エレメントに以下のオプションを使用したドリルダウンを実装する場合、このオプションは有効にしないでください。 • linkFields • linkSearch • linkView
field	フィールド名	返された最初のフィールド。	表示するフィールド。
link.exportResults.visible	論理値	(説明を参照)	パネルの下部に [エクスポート] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.inspectSearch.visible	論理値	(説明を参照)	パネルの下部に [調査] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.openPivot.visible	論理値	(説明を参照)	パネルの下部に [ピボットで開く] ボタンを表示します。 **デフォルト値**：link.visible の値。

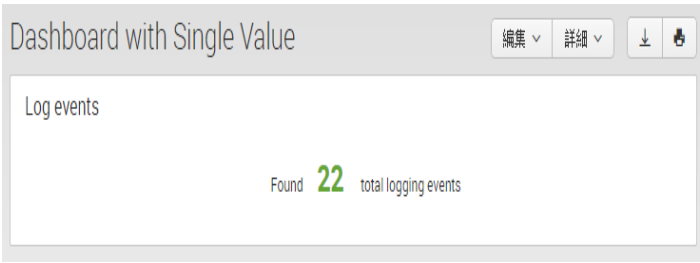
link.openSearch.search	サーチ文字列	—	[サーチで開く] ボタンで使用する代替サーチ。 link.openSearch.search で指定する代替サーチで使用する、もっとも早い時間。
link.openSearch.searchEarliestTime	(時間修飾子)	(説明を参照)	**デフォルト値**：パネルが使用するもっとも早い時間。 時間修飾子を使って時間を指定します。サーチの時間修飾子については、「時間修飾子の指定」を参照してください。
link.openSearch.searchLatestTime	(時間修飾子)	(説明を参照)	link.openSearch.search で指定する代替サーチで使用する、もっとも遅い時間。 **デフォルト値**：パネルが使用するもっとも遅い時間。 時間修飾子を使って時間を指定します。サーチの時間修飾子については、「時間修飾子の指定」を参照してください。
link.openSearch.text	テキスト	サーチで開く	[サーチで開く] ボタンで使用するラベル。
link.openSearch.ViewTarget	ビュー名	サーチ	[サーチで開く] ボタンのターゲットビュー。
link.openSearch.visible	論理値	(説明を参照)	パネルの下部に [サーチで開く] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.visible	論理値	false	パネルの下部にリンクボタンを表示します。
linkFields	(result | beforelabel | afterlabel | underlabel) カンマ区切りリスト	結果	単一値のどの部分のテキストをドリルダウンのリンクとして使用するかを設定します。結果と両方のラベルにリンクするための設定： `result, beforelabel, afterlabel` **注意**：ドリルダウンを有効にするには、linkFields および linkSearch プロパティの両方が必要です。linkView プロパティは省略することができます。ドリルダウン用サーチ以外のターゲットビューを指定するには、linkView を使用します。 linkFields、linkSearch、および linkView オプションを使用する場合は、<single> エレメントで **drilldown** オプションを有効にしないでください。
linkSearch	サーチ文字列		結果をクリック可能リンクに変換する、有効な完全サーチクエリー。 **注意**：ドリルダウンを有効にするには、linkFields および linkSearch プロパティの両方が必要です。linkView プロパティは省略することができます。ドリルダウン用サーチ以外のターゲットビューを指定するには、linkView を使用します。 linkFields、linkSearch、および linkView オプションを使用する場合は、<single> エレメントで **drilldown** オプションを有効にしないでください。
			ドリルダウンに対して、リンクされたサーチを実行するビューを指定します。 App がある任意のビューまたはグローバル権限がある任意のビューを指定することができます。

linkView	ビュー名	(説明を参照)	linkView のデフォルト値はありません。値を指定しない場合、ドリルダウン動作は無効になります。 linkView オプションを使用する場合は、<single> エレメントで drilldown オプションを有効にしないでください。
refresh.auto.interval	数値	0	更新間隔を秒で指定します。パネルの更新を無効にする場合は、0 (または負の整数) を指定します。
refresh.time.visible	論理値	true	パネルに更新時間インジケータを表示します。
refresh.link.visible	論理値	false	パネルに更新リンクを表示します。
underLabel	文字列		結果の下に表示するラベル。

例

ラベルの前後に表示する単一値パネル、および色範囲の指定例。サーチ内の範囲マップは、各範囲の値を指定します。このパネルは、範囲マップに対して Splunk のデフォルトの色を使用します。

```
<dashboard>
  <label>Dashboard with Single Value</label>
  <row>
    <panel>
      <single>
        <search>
          <query>
          index=_internal source="*splunkd.log" ( log_level=ERROR
          OR log_level=WARN* OR log_level=FATAL
          OR log_level=CRITICAL) | stats count as log_events
          | rangemap field=log_events low=1-100 elevated=101-300 default=severe
          </query>
          <earliest>-1d</earliest>
          <latest>now</latest>
        </search>
        <title>Log events</title>
        <option name="classField">range</option>
        <option name="afterLabel">total logging events</option>
        <option name="beforeLabel">Found</option>
      </single>
    </panel>
  </row>
</dashboard>
```



table

<table>

サーチデータをテーブルとして表示するパネル。

属性

名前	タイプ	デフォルト	説明
			この視覚エフェクトを表示するには、リストに記載されている 1 つまたは複数の

depends	トークンのカンマ区切りリスト	トークンが存在している必要があります。1 つまたは複数のトークンを指定することができます。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。
id	テキスト	視覚エフェクトの ID。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。 以下の用語は内部使用向けに予約されており、`id` に使用することはできません。 dashboard, search, default, submitted, footer, url, header
rejects	トークンのカンマ区切りリスト	この視覚エフェクトの表示を防止するには、このリスト内の 1 つ以上のトークンが存在している必要があります。 フォーム入力から、またはページ内ドリルダウンからのトークンを利用できます。

親エレメント

<row>
<panel>

```
<table>
  <title> (0..1)
  <search> (0..1)
    <earliest> (0..1)
    <latest> (0..1)
  <fields> (0..1)
  <drilldown> (0..n)
  <format type="sparkline" field="[field name]"> (0..n)
  <option name="[property]"> (0..n)
```

子エレメント

エレメント	タイプ	デフォルト	説明
<format>	テキスト		テーブル内にどのようにスパークラインを表示するかを決定する、一連の書式設定オプション。 詳細は、「スパークラインのオプション」を参照してください。

オプション

プロパティ	タイプ	デフォルト	説明
count	整数	10	表示する最大行数。
dataOverlayMode	(heatmap | highlow)	なし	表示するオーバーレイのタイプを示します。
displayRowNumbers	論理値	True	(廃止予定) rowNumbers 属性を使用します。
drilldown	(all | cell | row | none | off)	cell	行またはセルレベルでドリルダウンを有効にする、またはドリルダウンを無効にします。 **all**、**cell**：ドリルダウンを有効にします。これらの 2 つの値は同等です。セルレベルでドリルダウンを有効にします。 **row**：行のドリルダウンを有効にします。 **none**：ドリルダウンを無効にしますが、ハイパーテキストのスタイルは保持します。

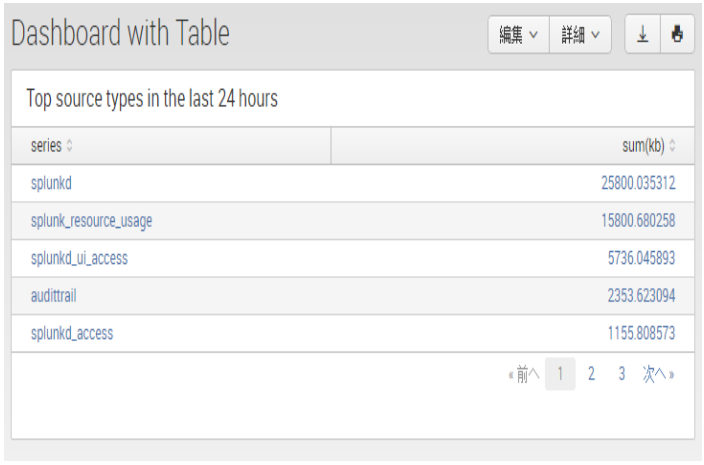
			off：ドリル ダウンを無効にして、ハイパーテキストのスタイルを削除します。
link.exportResults.visible	論理値	(説明を参照)	パネルの下部に [エクスポート] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.inspectSearch.visible	論理値	(説明を参照)	パネルの下部に [調査] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.openPivot.visible	論理値	(説明を参照)	パネルの下部に [ピボットで開く] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.openSearch.search	サーチ文字列	—	[サーチで開く] ボタンで使用する代替サーチ。
link.openSearch.searchEarliestTime	(時間修飾子)	(説明を参照)	link.openSearch.search で指定する代替サーチで使用する、もっとも早い時間。 **デフォルト値**：パネルが使用するもっとも早い時間。 時間修飾子を使って時間を指定します。サーチの時間修飾子については、「時間修飾子の指定」を参照してください。
link.openSearch.searchLatestTime	(時間修飾子)	(説明を参照)	link.openSearch.search で指定する代替サーチで使用する、もっとも遅い時間。 **デフォルト値**：パネルが使用するもっとも遅い時間。 時間修飾子を使って時間を指定します。サーチの時間修飾子については、「時間修飾子の指定」を参照してください。
link.openSearch.text	テキスト	サーチで開く	[サーチで開く] ボタンで使用するラベル。
link.openSearch.ViewTarget	ビュー名	サーチ	[サーチで開く] ボタンのターゲットビュー。
link.openSearch.visible	論理値	(説明を参照)	パネルの下部に [サーチで開く] ボタンを表示します。 **デフォルト値**：link.visible の値。
link.visible	論理値	true	パネルの下部にリンクボタンを表示します。
previewResults	論理値	True	サーチ完了前の結果のプレビューを有効にします。
refresh.auto.interval	数値	0	更新間隔を秒で指定します。パネルの更新を無効にする場合は、0 (または負の整数) を指定します。
refresh.time.visible	論理値	true	パネルに更新時間インジケータを表示します。
refresh.link.visible	論理値	true	パネルに更新リンクを表示します。
rowNumbers	論理値	False	行番号の表示を切り替えます。
showPager	論理値	True	ページャーをオン/オフにします。
wrap	論理値	True	結果テーブル内のテキストの折り返し表示を有効にします。

例

行番号を省略し、5 行を表示するインラインサーチを使用したテーブルパネルの例：

```
<dashboard>
```

```
  <label>Dashboard with Table</label>
  <row>
    <panel>
      <table>
        <title>Top source types in the last 24 hours</title>
        <search>
          <query>
          index=_internal group=per_sourcetype_thruput
          | chart sum(kb) by series | sort -sum(kb)
          </query>
          <earliest>-24h</earliest>
          <latest>now</latest>
        </search>
        <option name="count">5</option>
        <option name="rowNumbers">0</option>
      </table>
    </panel>
  </row>
</dashboard>
```



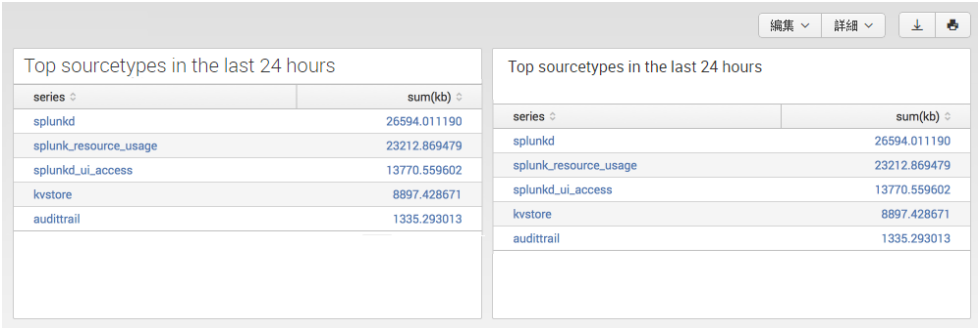
タイトル

<タイトル>
<panel> エレメントのタイトル・テキストまたは視覚化エレメントのタイトルを指定します。
親エレメント <panel> <chart> | <event> | <html> | <map> | <single> | <table>
<panel> <title> (0..1) <!-- Title at panel level --> <chart> | <event> | <html> | <map> | <single> | <table> (1..n) <title> (0..1) <!-- Title at visualization level -->
属性 <title> の属性はありません。>
例 <table> 視覚エフェクトを含む <panel> のタイトルを指定します。 <panel> <title>Top sourcetypes in the last 24 hours</title> <table> <search> <query> index=_internal group=per_sourcetype_thruput

```
      | chart su(kb) by series | sort -sum(kb)
      </query>
      <earliest>-24h</earliest>
      <latest>now</latest>
    </search>
    <option name="count">5</option>
    <option name="rowNumbers">0</option>
  </table>
</panel>
```

<table> 視覚エフェクトのタイトルを指定します。

```
<panel>
  <table>
    <title>Top sourcetypes in the last 24 hours</title>
    <search>
      <query>
        index=_internal group=per_sourcetype_thruput
        | chart su(kb) by series | sort -sum(kb)
      </query>
      <earliest>-24h</earliest>
      <latest>now</latest>
    </search>
    <option name="count">5</option>
    <option name="rowNumbers">0</option>
  </table>
</panel>
```



Sparkline のオプション

<format type="sparkline" field="[field name]">			
属性			
名前	**タイプ**	**デフォルト**	**説明**
field	フィールド名		**必須。** スパークラインを適用するフィールドを指定します。
type	文字列	sparkline	**必須。** `sparkline`が唯一サポートされているタイプです。スパークラインが書式設定中かどうかを示します。

テーブル内にどのようにスパークラインを表示するかを決定する、一連の書式設定オプション。スパークラインのオプションは、<table> エレメントにのみ適用されます。<table> エレメント内に <format> エレメントを使って、sparkline オプションを指定します。

スパークラインの書式を設定するここのスパークラインオプションと、chart または stats サーチコマンドの sparkline 関数を混同しないでください。ここに記載している書式設定オプションには、sparkline() 関数を使用するサーチが必要です。スパークラインの実装については、「サーチ結果へのスパークラインの追加」を参照してください。

警告：このリファレンスに記載されているスパークラインオプションは、ダッシュボードの PDF 生成時には表示されません。スパークライン自身のみが表示されます。

親エレメント

<table>

```
<table>
   <format type="sparkline" field=["field name]"> (0..n)
     <option name="[property name]"> (0..n)
```

共通オプション

プロパティ	タイプ	デフォルト	説明
height	CSS スタイル	auto	グラフの高さ。有効な CSS 幅を指定します (例：1.5em<20px)。
tooltipPrefix	テキスト		ツールヒントに表示される各フィールドの前に表示するテキスト。
tooltipSuffix	テキスト		ツールヒントに表示される各フィールドに追加されるテキスト。
type	(bar | discrete | line)	line	スパークラインのタイプを示します。

横棒グラフ (bar) のオプション

プロパティ	タイプ	デフォルト	説明
barSpacing	数値		各横棒間の間隔 (ピクセル)。
barWidth	数値		各横棒の幅 (ピクセル)。
colorMap	説明を参照		特定の値を選択した色とマップする範囲マップ。 たとえば、値が -2 のすべての値を黄色で表示する場合、次の colorMap を使用します：{ '-2':'#ff0' }。 ここで、各横棒の色のマッピングを個別に指定する代わりに、値の配列を渡すことができます。たとえばグラフに3つの値 1,3,1 がある場合、colorMap=["red", "green", "blue"] と設定することができます。

離散型グラフ (discrete) のオプション

プロパティ	タイプ	デフォルト	説明
lineColor	CSS スタイル		折れ線および離散型グラフで、CSS 値文字列として描画される線の色を指定するために使用します。
lineHeight	数値	グラフの高さの 30%	各折れ線の高さ (ピクセル)。
thresholdColor	CSS の色		thresholdValue と組み合わせて使用する CSS の色。
thresholdValue	CSS の色		lineColor の代わりに thresholdColor を使って、これ未満の値を描画します。

折れ線グラフ (line) のオプション

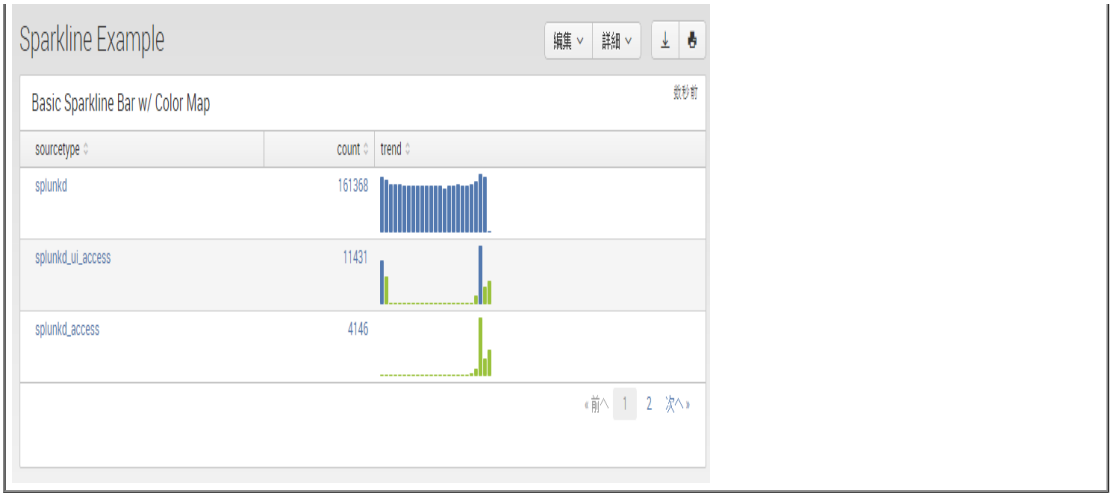
プロパティ	タイプ	デフォルト	説明
fillColor	CSS の色 | false		グラフ下の領域に CSS 値として塗りつぶす色を指定します。塗りつぶしを無効にする場合は false を設定します。
highlightLineColor	CSS の色	#f22	マウスカーソルをかざした時に、値を介して表示される垂直線の CSS 色。 無効にする場合は NULL を設定します。
highlightSpotColor	CSS の色	#f5f	マウスカーソルをかざした時に、値上に表示されるスポットの CSS 色。 無効にする場合は NULL を設定します。
lineColor	CSS スタイル		折れ線および離散型グラフで、CSS 値文字列として描画される線の色を指定するために使用します。
lineWidth	数値	1	線の幅 (ピクセル)。
maxSpotColor	CSS の色		最大値に対して表示されるマーカーの CSS 色。 非表示にするには、false または空文字列を設定します。

minSpotColor	CSS の色		最小値に対して表示されるマーカーの CSS 色。 非表示にするには、false または空文字列を設定します。
normalRangeMax	範囲 (説明を参照)		normalRangeMin とともに、これらの閾値の範囲内に該当する場合、「正常」または予期する値の範囲であることを表す横棒を描画します。 たとえば、この範囲 80,85,84,88,98,114,116,104,95,85,84 で緑 (正常) の横棒が、正常な動作温度範囲を表します。
normalRangeMin	範囲 (説明を参照)		normalRangeMax とともに、これらの閾値の範囲内に該当する場合、「正常」または予期する値の範囲であることを表す横棒を描画します。 たとえば、この範囲 80,85,84,88,98,114,116,104,95,85,84 で緑 (正常) の横棒が、正常な動作温度範囲を表します。
spotColor	CSS の色		最後の値マーカーの CSS 色。 非表示にするには、false または空文字列を設定します。
spotRadius	数値	1.5	すべてのスポットマーカーの半径 (ピクセル)。
valueSpots	範囲 (説明を参照)		スポットの描画場所、および描画色。範囲を受け付けます。 たとえば、50 未満のすべての値に緑のスポットを、それ以上の値に赤のスポットを表示する場合は、{':49': 'green, '50:': 'red'} を使用します。
width	CSS スタイル	auto	グラフの幅。グラフの高さ。有効な CSS 幅を指定します (例：1.5em<20px)。このオプションは、bar および tristate タイプのスパークラインに適用されます。

例

タイプが bar で色マップを持つスパークライン

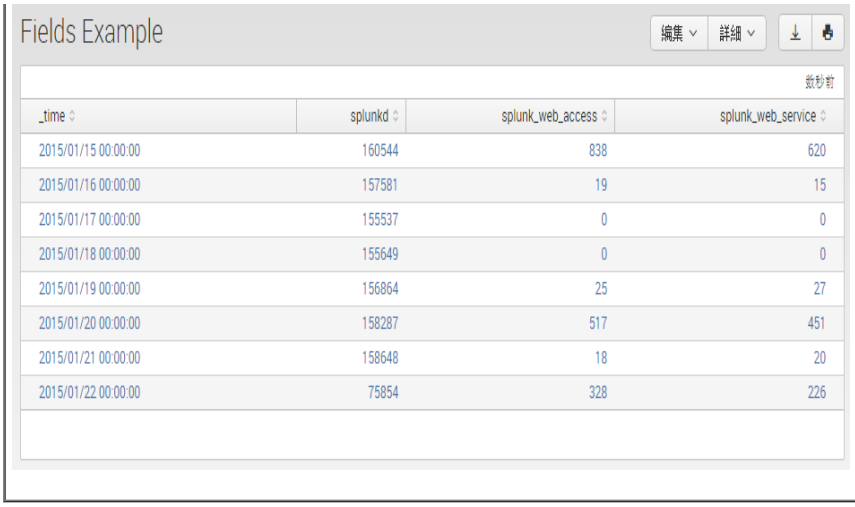
```
<dashboard>
  <label>Sparkline Example</label>
  <row>
    <panel>
      <table>
        <title>Basic Sparkline Bar w/ Color Map</title>
        <!-- Set span for each sparkline datapoint to be 1 hour -->
        <search>
          <query>
          index=_internal | chart count sparkline(count, 1h) as trend by sourcetype | sort -count
          </query>
          <earliest>-24h@h</earliest>
          <latest>now</latest>
        </search>
        <option name="count">3</option>

        <!-- Set sparkline options here; make sure that field matches field name of the search results -->
        <format type="sparkline" field="trend">
          <option name="type">bar</option>
          <option name="height">40px</option>
          <!-- Use colorMap to map specific values to selected colors -->
          <option name="colorMap">
            <option name="2000:">#5379AF</option>
            <option name=":1999">#9ac23c</option>
          </option>
          <option name="barWidth">5px</option>
        </format>
      </table>
    </panel>
  </row>
</dashboard>
```



フィールド

<フィールド>
カンマ区切り形式のフィールド。サーチをこれらのフィールドに制限するには、<fields> エレメントを使用します。 カンマ区切りリスト内のフィールドの順序により、テーブル内の列またはイベントの表示順序が決まります。
親エレメント 　　<event> <table>
`<event> | <table>` `  <fields> (0..1)`
例 サーチの結果を以下のフィールドに制限します。 　　_time、splunkd、splunk_web_access、splunk_web_service

```
<dashboard>
  <label>Fields Example</label>
  <row>
    <panel>
      <table>
        <search>
          <query>
          index=_internal | timechart count by sourcetype
          </query>
          <earliest>-7d@d</earliest>
          <latest>now</latest>
        </search>
        <fields>_time, splunkd, splunk_web_access, splunk_web_service</fields>
        <option name="rowNumbers">0</option>
      </table>
    </panel>
  </row>
</dashboard>
```



_time	splunkd	splunk_web_access	splunk_web_service
2015/01/15 00:00:00	160544	838	620
2015/01/16 00:00:00	157581	19	15
2015/01/17 00:00:00	155537	0	0
2015/01/18 00:00:00	155649	0	0
2015/01/19 00:00:00	156864	25	27
2015/01/20 00:00:00	158287	517	451
2015/01/21 00:00:00	158648	18	20
2015/01/22 00:00:00	75854	328	226

オプション

<option>

<option> タグは、エレメントの特定のプロパティに適用されます (panel エレメントなど)。プロパティを指定するには、name 属性を使用します。

一般的に名前付きオプションは特定のパネルに適用されます。ただし、一部のオプションは複数のパネルに適用することができます。

属性

名前	タイプ	デフォルト	説明
name	プロパティ名 (必須)		特定のプロパティの名前を指定します。 <option> に指定できる値は、名前付きプロパティによって異なります。名前付きオプションと利用できる値のリストについては、各パネルの参照エントリを参照してください。

親エレメント

<chart> <event> <single> <table>

```
<chart> |  <event> | <html> | <single> | <table>
  . . .
  <option name="[property]">[option value]</option> (0..n)
```

例

```
<table>
  <title>Top sourcetypes in the last 24 hours</title>
  <searchString>
    index=_internal group=per_sourcetype_thruput | chart sum(kb) by series | sort -sum(kb)
  </searchString>
  <earliestTime>-1d</earliestTime>
  <latestTime>now</latestTime>
  <option name="count">5</option>
  <option name="rowNumbers">0</option>
</table>
```

search エレメント

<dashboard>、<form>、および panel 視覚化エレメントのサーチを指定するには、<search> エレメントを使用します。フォーム入力の選択項目の設定にも、<search> エレメントを使用します。

search

<search>

ダッシュボード、フォーム、またはパネルのサーチを定義します。フォーム入力の場合、入力の選択項目を動的に定義します。

インライン・サーチ
視覚エフェクトに指定されているサーチ。インライン・サーチを指定するには、<query> エレメントを使用します。

レポート
レポートから参照されたサーチ。レポートを参照するには、`ref` 属性を使用します。パネルには、参照先レポートのサーチと視覚エフェクトの両方に基づく視覚エフェクトが含まれます。サーチを変更することはできませんが、サーチ結果の視覚エフェクトを変更、設定することは可能です。レポート内のサーチが変更されると、そのレポートに基づくパネルにも変更が反映されます。

入力の設定用サーチ
フォーム入力の選択項目を設定するサーチ。チェックボックス、ドロップダウン、複数選択、およびラジオ入力の選択項目を設定するには、<search> をフォーム入力の子エレメントとして使用します。設定用サーチは選択項目の設定に、フォーム入力属性 `fieldForLabel` および `fieldForValue` を使用します。設定用サーチにリアルタイム・サーチは使用しないでください。リアルタイム・サーチを使用すると、入力用の選択項目が正しく更新されません。

グローバル・サーチ
<dashboard> または <form> コンテキストからのサーチは、グローバル・サーチです。グローバル・サーチは後処理サーチのベース・サーチとして使用します。後処理サーチが参照できるように、グローバル・サーチには常に `id` 属性が必要です。

後処理サーチ
ベース・サーチの結果をさらに変更するサーチ。後処理サーチを導入するには、`base` および `id` 属性を使用します。後処理サーチは `base` 属性を使ってベース <search> の `id` 属性を参照します。

ベース・サーチは、グローバル・サーチまたはパネル・レベルのサーチです。ベース・サーチには、<earliest> および <latest> エレメントを指定します。後処理サーチは、それの子エレメントとなる <earliest> および <latest> エレメントを無視します。

> **警告**：後処理サーチに対して、10,000 件を超えるイベントを返すベース・サーチは使用しないでください。イベント・ベースのサーチの場合、ベース・サーチが後処理サーチに渡せるのは最大で 10,000 件の raw イベントであるという、変更不可能な制限があります。後処理サーチは、この 10,000 件のイベント制限を超えたイベントを処理せず、単純に無視します。そのため後処理サーチのデータは不完全になります。これは、結果ベースのサーチには適用されません。
>
> **警告**：ベースサーチから大量のサーチ結果を渡すと、サーバータイムアウトが発生する可能性があります。このような場合は、以下の事項を検討してください。
>
> - ベースサーチが返す結果数とフィールド数。
> - これらの結果に対する後処理操作の複雑性。
>
> 後処理サーチの詳細は、「後処理サーチ」を参照してください。

属性

名前	タイプ	デフォルト	説明
app	テキスト		App 名。 現在の App にないレポートを参照するには、`app` 属性と `ref` 属性を使用します。
base	テキスト		後処理サーチによる、ベース・サーチへの参照。 サーチの `id` 属性で、現在のダッシュボードのベース・サーチを参照してください。
id	テキスト		サーチの ID。後処理サーチは、この ID を使ってベース・サーチを参照します。 英数字とアンダースコアのみ使用できます。数字またはアンダースコアで開始することはできません。 以下の用語は内部使用向けに予約されており、`id` に使用することはできません。 dashboard, search, default, submitted, footer, url, header
ref	テキスト		サーチを含んでいるレポートへの参照。 他の App のレポートを参照する場合は、`app` 属性を使って App を指定します。

親エレメント

<form>
<dashboard>
<panel>
<chart> <event> <map> <single> <table>

子エレメント

エレメント	タイプ	デフォルト	説明
<query>	テキスト		クエリのサーチ文字列。
<earliest> <latest>	テキスト		もっとも早い/もっとも遅い時間パラメータを示す時間式 (省略可)。 後処理サーチは、子エレメントの <earliest> および <latest> を無視します。代わりに、参照しているベース・サーチの <earliest> および <latest> エレメントを使用します。 時間は相対時間または絶対時間で指定できます。相対時間の場合、相対時間修飾子を使用します。「サーチへの時間修飾子の指定」を参照してください。絶対時間の場合、時間は UNIX エポック時フォーマットで指定します。

インライン・サーチからのベース・サーチ

```
<search id=[base ID]>
  <query>[search string]</query> (1)
  <earliest> (0..1)
  <latest> (0..1)
```

レポートからのベース・サーチ

```
<search id=[base ID] [ref=[report name]]>
  <earliest> (0..1)
  <latest> (0..1)
```

後処理サーチ

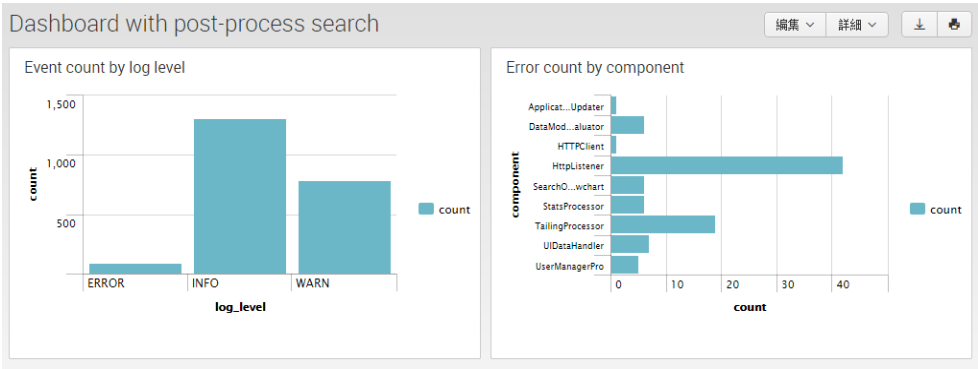
```
<search base=[base ID]> (0..n)
  <query>[post-process search string]</query> (1)
```

例

ベース・サーチと 2 つの後処理サーチを持つダッシュボード。

```
<dashboard>
  <label>Dashboard with post-process search</label>
  <description></description>
  <!-- Base search cannot pass more than 10,000 events to post-process searches-->
  <!-- Example uses stats transforming command -->
  <!-- This limits events passed to post-process search -->
  <search id="baseSearch">
    <query>
      index=_internal source=*splunkd.log | stats count by component, log_level
    </query>
    <earliest>-30d</earliest>
    <latest>now</latest>
  </search>
 <row>
    <panel>
      <chart>
        <title>Event count by log level</title>
        <!-- post-process search -->
        <search base="baseSearch">
          <query>
            stats sum(count) AS count by log_level
          </query>
        </search>
      </chart>
    </panel>
```

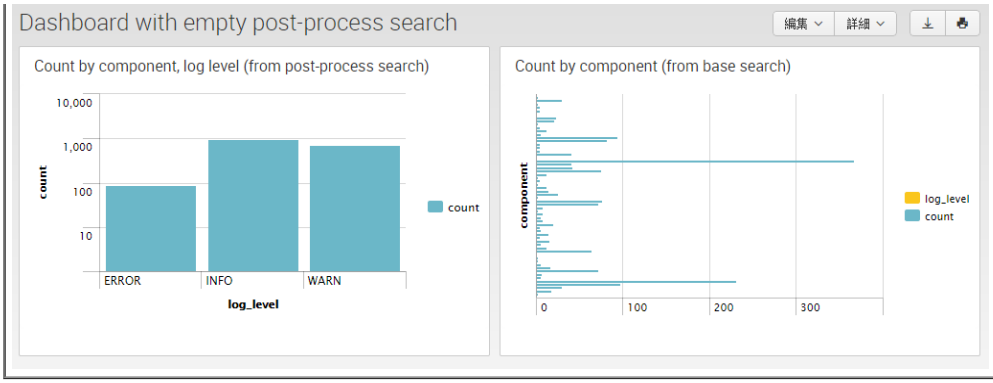
```
    <panel>
      <chart>
        <title>Error count by component</title>
        <!-- post-process search -->
        <search base="baseSearch">
          <query>
            search log_level=error | stats sum(count) AS count by component
          </query>
        </search>
        <option name="charting.chart">bar</option>
      </chart>
    </panel>
  </row>
</dashboard>
```



例

空の後処理サーチを使用するダッシュボード。

```
<dashboard>
  <label>Dashboard with empty post-process search</label>
  <description></description>
  <!-- Base search cannot pass more than 10,000 events to post-process searches-->
  <!-- Example uses stats transforming command -->
  <!-- This limits events passed to post-process search -->
  <search id="baseSearch">
    <query>index=_internal source=*splunkd.log | stats count by component, log_level</query>
    <earliest>-30d</earliest>
    <latest>now</latest>
  </search>
  <row>
    <panel>
      <chart>
        <title>Count by component, log level (from post-process search)</title>
        <!-- post-process search -->
        <search base="baseSearch">
          <query>stats sum(count) AS count by log_level</query>
        </search>
        <option name="charting.axisY.scale">log</option>
      </chart>
    </panel>
    <panel>
      <chart>
        <title>Count by component (from base search)</title>
        <!-- empty post-process search -->
        <search base="baseSearch" />
        <option name="charting.chart">bar</option>
      </chart>
    </panel>
  </row>
</dashboard>
```



サーチ・エレメント (廃止予定)

populatingSavedSearch

<populatingSavedSearch>
廃止予定：フォーム入力の選択項目を動的に設定するには、input エレメントの子エレメントとして <search> エレメントを使用します。 次の入力のオプションに対して、ラベルと対応する値を設定する、レポートからのサーチ： • <checkbox> • <dropdown> • <multiselect> • <radio> **警告**：設定用サーチにリアルタイム・サーチは使用しないでください。リアルタイム・サーチを使用すると、入力用の選択項目が正しく更新されません。 Splunk Web の入力エディタの **[動的]** セクションから、設定用の保存済みサーチを指定することができます。例については、「動的オプションによる選択項目の指定」を参照してください。この例はインラインサーチ用ですが、手順は同じです。

属性

名前	タイプ	デフォルト	説明
fieldForLabel	フィールド名		**必須**：サーチから生成された値のラベルに使用するフィールド。
fieldForValue	フィールド名		**必須**。サーチから生成された値の値に使用するフィールド。

親エレメント

<input type="radio"> | <input type="dropdown">

```
<populatingSavedSearch fieldForValue="[field name]" fieldForLabel="[field name]">
    [report name]
</populatingSavedSearch>
```

例

ドロップダウン入力を、ソースタイプ選択用動的設定オプションで設定します。「対応する <populatingSearch> の例」を参照してください。

```
<form>
  <label>Form Input Example (Populating Search)</label>
  <description/>
  <fieldset submitButton="false">
    <input type="dropdown" token="source_tok" searchWhenChanged="true">
      <label>Select a source type</label>
      <choice value="*">All</choice>
      <populatingSearch earliest="-24h@h" latest="now"
            fieldForLabel="sourcetype" fieldForValue="sourcetype">
               myReport
      </populatingSearch>
      <prefix>sourcetype="</prefix>
```

```
      <suffix>"</suffix>
      <default>*</default>
    </input>
  </fieldset>
  <row>
    <panel>
      <chart>
        <title>Source type count for last 7 days</title>
        <searchString>
          index=_internal $source_tok$ | timechart count
        </searchString>
        <earliestTime>-7d@h</earliestTime>
        <latestTime>now</latestTime>
        <option name="charting.chart">column</option>
      </chart>
    </panel>
  </row>
</form>
```

populatingSearch

<populatingSearch>

廃止予定：フォーム入力の選択項目を動的に設定するには、input エレメントの子エレメントとして <search> エレメントを使用します。

次の入力のオプションに対して、ラベルと対応する値を設定する、インラインサーチ：

- <checkbox>
- <dropdown>
- <multiselect>
- <radio>

警告：設定用サーチにリアルタイム・サーチは使用しないでください。リアルタイム・サーチを使用すると、入力用の選択項目が正しく更新されません。

Splunk Web の入力エディタの **[動的]** セクションから、設定用のサーチを指定することができます。例については、「動的オプションによる選択項目の指定」を参照してください。

属性

名前	タイプ	デフォルト	説明
fieldForLabel	フィールド名		**必須**：設定用サーチから生成された値のラベルに使用するフィールド。
fieldForValue	フィールド名		**必須**。設定用サーチから生成された値の値に使用するフィールド。
earliest latest	時間修飾子		サーチ結果を一定期間に制限します。earliest と latest の片方または両方を指定します。例：`earliest="-7d" latest="-1d"` と指定します。リアルタイムサーチを有効にする場合は、`rt` を指定します。時間修飾子の指定の詳細は、「<earliestTime>」および「<latestTime>」を参照してください。

親エレメント

<input type="radio"> | <input type="dropdown">

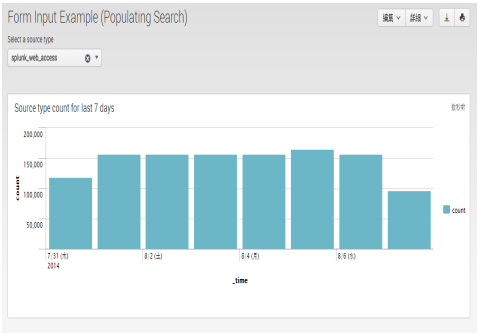
```
<populatingSearch
  fieldForValue="[field name]" fieldForLabel="[field name]"
  earliest="[timeformat]" latest="[timeformat]">
    [inline search]
</populatingSearch>
```

例

ドロップダウン入力を、ソースタイプ選択用動的設定オプションで設定します。

```
<form>
  <label>Form Input Example (Populating Search)</label>
  <description/>
  <fieldset submitButton="false">
    <input type="dropdown" token="source_tok" searchWhenChanged="true">
```

```
      <label>Select a source type</label>
      <choice value="*">All</choice>
      <populatingSearch earliest="-24h@h" latest="now"
            fieldForLabel="sourcetype" fieldForValue="sourcetype">
               index=_internal | stats count by sourcetype
      </populatingSearch>
      <prefix>sourcetype="</prefix>
      <suffix>"</suffix>
      <default>*</default>
    </input>
  </fieldset>
  <row>
    <panel>
      <chart>
        <title>Source type count for last 7 days</title>
        <searchString>
          index=_internal $source_tok$ | timechart count
        </searchString>
        <earliestTime>-7d@h</earliestTime>
        <latestTime>now</latestTime>
        <option name="charting.chart">column</option>
      </chart>
    </panel>
  </row>
</form>
```



Viz pop search example2.png

searchName

<searchName>
廃止予定：サーチを含むレポートを参照するには、<search> エレメントと <name> エレメントを使用します。 パネルが使用するサーチを含むレポート名。
親エレメント <chart> <event> <list> <map> <single> <table>
`<searchName>[report name]</searchName>`

例 (パネル)
`<chart>`
`  <searchName>Splunk errors last 24 hours</searchName>`
`</chart>`

searchString

<searchString>
廃止予定：サーチを定義するには、<search> エレメントを使用します。 視覚エフェクトエレメントが結果の表示に使用するインラインサーチ。

親エレメント

<chart> <event> <list> <map> <single> <table>

```
<chart> |  <event> | <html> | <list> | <map> | <single> | <table>
  <searchString>[search]</searchString> (1)
  <earliestTime>[relative time expression></earliestTime> (0..1)
  <latestTime>[relative time expression></latestTime> (0..1)
```

属性

<searchString> の属性はありません。

例 (パネル)

```
<table>
  <searchString>
      index="_internal" source="*metrics.log" group="pipeline"
      | chart sum(cpu_seconds) over processor | sort -sum(cpu_seconds)
      |rename sum(cpu_seconds) as "Total CPU Seconds"
  </searchString>
  <title>High CPU processors</title>
  . . .
</table>
```

例 (フォーム)

```
<form>
  <fieldset>
   <input type="text" token="sourcetype" />
  </fieldset>
  <searchString>
    index=_internal source=*metrics.log group=per_sourcetype_thruput
      series="$sourcetype$" | head 1000
  </searchString>
  <row>
    <panel>
      <table>
        <title>Matching events</title>
        <option name="count">50</option>
      </table>
    </panel>
  </row>
</form>
```

searchPostProcess

<searchPostProcess>

廃止予定：ベース・サーチと後処理用サーチを定義するには、<search> エレメントと `id` および `ref` 属性を使用します。

パネル内のベースサーチからのイベントまたは結果を処理するインラインサーチ文字列。一般的にベースサーチは、変換サーチになります。

警告：後処理サーチには、渡して処理できる raw イベントの件数が 10,000 件に制限されており、この制限値を変更することはできません。10,000 件の制限を超えたイベントは処理されず、無視されます。また、無視された場合に何のメッセージも表示されないため、後処理サーチで報告されたデータが不完全になる可能性があります。

警告：ベースサーチから大量のサーチ結果を渡すと、サーバータイムアウトが発生する可能性があります。このような場合は、以下の事項を検討してください。

- ベースサーチが返す結果数とフィールド数。
- これらの結果に対する後処理操作の複雑さ。

後処理サーチの詳細は、「Use one search for a whole dashboard」を参照してください。このトピックは、アドバンスト XML マニュアルに記載されていますが、シンプル XML の後処理サーチにも原則は適用されます。

親エレメント

<chart> <event> <list> <single> <table>

```
<searchPostProcess>[search string]</searchPostProcess>
```

例

```
<form>
  <fieldset>
      <input type="dropdown" token="reportTypeToken">
            <label>Select name</label>
            <default>Sourcetype</default>
            <choice value="index">Index</choice>
            <choice value="sourcetype">Sourcetype</choice>
            <choice value="source">Source</choice>
            <choice value="host">Host</choice>
        </input>
      <input type="time">
        <default>Last 4 hours</default>
      </input>
  </fieldset>

  <!-- Search that returns all of the data that requested by subsequent panels -->
  <searchTemplate>
    index=_internal source=*metrics.log group="per_$reportTypeToken$_thruput"
    | bin _time span=1m | stats count by series, eps, kb, kbps, _time
  </searchTemplate>

  <row>
    <panel>
      <table>
          <title>eps over time</title>
          <searchPostProcess>timechart avg(eps) by series</searchPostProcess>
      </table>
    </panel>
    <panel>
      <chart>
          <title>KB indexed over time</title>
          <searchPostProcess>timechart sum(kb) by series</searchPostProcess>
          <option name="height">300px</option>
          <option name="charting.chart">area</option>
          <option name="charting.chart.stackMode">stacked</option>
      </chart>
    </panel>
  </row>
</form>
```

searchTemplate

<searchTemplate>
廃止予定：フォームからのユーザー入力を置換するトークンを区切るためには、$*token*$ を使用するベース・サーチで <search> エレメントを使用します。 $ トークン$ を使って、フォームからのユーザー入力と置換するトークンを区切る、フォームのベースサーチ。 <searchTemplate> は、<dashboard> または panel で使用することもできます。
親エレメント <form> <dashboard> <chart> | <event> | <html> | <list> | <single> | <table>

例

```
<form>
  <label>Basic form search</label>
  <fieldset>
    <html>
      <p>
       Enter a sourcetype in the field below.
      </p>
    </html>
     <!-- the default input type is a text box -->
     <input token="sourcetype" />
  </fieldset>
  <!-- search with replacement token delimited with $ -->
```

```
  <searchTemplate>
   index=_internal source=*metrics.log
     group=per_sourcetype_thruput series="$sourcetype$"
     | head 1000
  </searchTemplate>
  <row>
    <panel>
      <!-- output the results as a 50 row events table -->
      <table>
        <title>Matching events</title>
        <option name="count">50</option>
      </table>
    </panel>
  </row>
</form>
```

earliestTime

<earliestTime>
廃止予定：サーチに含めるもっとも早い時間を指定するには、<earliest> エレメントを使用します。 サーチに含めるもっとも早い時間を指定します。 時間は相対時間または絶対時間で指定できます。相対時間の場合、相対時間修飾子を使用します。「サーチへの時間修飾子の指定」を参照してください。絶対時間の場合、時間は UNIX エポック時フォーマットで指定します。 パネル視覚エフェクトエレメントの子として指定された場合は、その視覚エフェクトの時間を変更します。ダッシュボードまたはフォームに対して指定された場合、ダッシュボードまたはフォームのサーチを変更します。
親エレメント <form> | <dashboard> <chart> <event><list> <single> <table>
`<earliestTime>[time modifier]</earliestTime>`

例

フォーム内の 1 つの視覚エフェクトの時間範囲を変更します。

```
<form>
  <searchTemplate>
    index=_internal source=*metrics.log group="per_sourcetype_thruput"
    | fields eps, kb, kbps
  </searchTemplate>
  . . .
  <row>
    <panel>
      <table>
        <title>Last Seven Days</title>
        <earliestTime>-7d</earliestTime>
        <latestTime>now</latestTime>
      </table>
    </panel>
  </row>
</form>
```

ダッシュボード内のグラフの時間範囲を変更します。

```
<dashboard>
  <row>
    <panel>
      <chart>
        <title>Error log count</title>
        <option name="charting.chart">radialGauge</option>
        <searchString>
           index=_internal source="*splunkd.log"
           ( log_level=ERROR OR log_level=WARN*
           OR log_level=FATAL OR log_level=CRITICAL )
```

```
          | stats count as log_events
          | rangemap field=log_events low=1-100 elevated=101-300 default=severe
        </searchString>
        <earliestTime>-24h@h</earliestTime>
        <latestTime>now</latestTime>
      </chart>
    <panel>
  </row>
</dashboard>
```

latestTime

<latestTime>
廃止予定：サーチに含めるもっとも遅い時間を指定するには、<latest> エレメントを使用します。 サーチに含めるもっとも遅い時間を指定します。 時間は相対時間または絶対時間で指定できます。相対時間の場合、相対時間修飾子を使用します。「サーチへの時間修飾子の指定」を参照してください。絶対時間の場合、時間は UNIX エポック時フォーマットで指定します。 パネルエレメントの子として指定された場合は、そのパネルの時間を変更します。ダッシュボードまたはフォームに対して指定された場合、ダッシュボードまたはフォームのサーチを変更します。
親エレメント <form> | <dashboard> <chart> <event><list> <single> <table>
<latestTime>[time modifier]</latestTime>

例
`<form>`
`  <searchTemplate>`
`  index=_internal source=*metrics.log group="per_sourcetype_thruput"`
`  | fields eps, kb, kbps`
`  </searchTemplate>`
`  . . .`
`  <row>`
`  <panel>`
`  <table>`
`  <title></title>`
`  <earliestTime>-7d</earliestTime>`
`  <latestTime>now</latestTime>`
`  </table>`
`  </panel>`
`  </row>`
`</form>`

ドリル ダウンエレメント

drilldown

<drilldown>
ユーザーがダッシュボードやフォーム内のフィールドをクリックした時のカスタム宛先リンクを定義します。 <link> タグを使って、宛先へのパスを指定します。 <set> または <unset> タグを使って、トークンを設定/解除します。 トークンを設定/解除するフィールドを指定する条件を指定します。 **注意：**1 つまたは複数のアクション (<link>、<set>、<unset>) または条件 (<condition>) を直接 <drilldown> 内に指定できますが、アクションと条件の両方を指定することはできません。 詳細は、「ダッシュボードとフォームの動的なドリルダウン」を参照してください。

属性

名前	タイプ	デフォルト	説明

target	テキスト	—	<a> HTTP タグの target 属性に対応しています。 ドリルダウンを新しいウィンドウで表示するには、「_blank」を指定します。 ドリルダウンを同じウィンドウで表示するには、「_self」を指定します。 ドリルダウンを新しいウィンドウで表示するには、任意の文字列を指定します。このターゲットへのそれ以降の参照は、このウィンドウ内に開かれます。

親エレメント

<chart> <event> <map> <single> <table>

```
<drilldown>
  ( <link> | <set>  | <unset> ) (1..n) | <condition> (1..n)
```

例 1：フォームに値を渡す

```
<table>
<searchString>index=_internal</searchString>

<!-- Pass the clicked row's 'count'-column value    -->
<!-- to populate a destination form's 'foo' token. -->
<drilldown>
  <link>
  /app/search/simple_xml_form?form.foo=$row.count$
  </link>
</drilldown>
</table>
```

例 2：フォームにパラメータを渡す

```
<table>
<searchString>index=_internal</searchString>

<!-- Pass the clicked cell's value, earliest time, -->
<!-- and latest time to a destination form's       -->
<!-- token ('foo') and search parameters           -->
<drilldown>
  <link>
  <![CDATA[
/app/search/simple_xml_form?form.foo=$click.value2$&earliest=$earliest$&latest=$latest$
  ]]>
  </link>
</drilldown>
</table>
```

例 3：グラフの値を Web サイトに渡す

```
<chart>
  <searchString>
    index=_internal | chart count by sourcetype
  </searchString>
  <option name="charting.chart">column</option>

  <!-- $click.value$ captures the value clicked by the user -->
  <!-- From the x-axis of a column chart and passes         -->
  <!-- it to the website as a query parameter               -->
  <drilldown>
    <link>
      http://splunk-base.splunk.com/integrated_search/?q=$click.value$
    </link>
  </drilldown>
</chart>
```

condition (drilldown)

<condition>

ドリルダウンアクションの範囲を、特定のフィールドのクリックに制限します。<condition> エレメントが存在しない場合は、すべてのフィールドにドリルダウンアクションが適用されます。

注意：<condition> エレメントは、input エレメントと drilldown エレメントの両方に適用されます。詳細は「<condition> (input)」を参照してください。

親エレメント

<drilldown>

```
<condition>
  (<link> | <set> | <unset>) (1..n)
```

属性

名前	タイプ	デフォルト	説明
field	テキスト	*	ドリルダウンを実装する、またはトークンを設定/解除するサーチフィールドを指定します。
label	テキスト	*	**入力コンテキストのみ。**条件を適用する入力 <label> エレメントを指定します。 「*」の場合、すべての入力 <label> エレメントに条件を適用します。 「<condition> (input)」を参照してください。
値	テキスト	*	**入力コンテキストのみ。**条件を適用する入力 <value> エレメントを指定します。 「*」の場合、すべての入力 <value> エレメントに条件を適用します。 「<condition> (input)」を参照してください。

例

<condition> タグを使ったページ内ドリルダウン用トークンの設定については、<set> の例を参照してください。

複数の <condition> タグの使用については、<unset> の例を参照してください。

selection

<selection>

グラフのパン/ズーム機能の時間ウィンドウを設定します。また、トークンを使って、グラフの X 軸の数値などの、その他の値を設定することもできます。

タイプが area、column、または line のグラフにのみ適用されます。

グラフのパン/ズーム機能の詳細は、「グラフのコントロール」を参照してください。

親エレメント

```
<chart>
  <option name="charting.chart">area</option>
  | <option name="charting.chart">column</option>
  | <option name="charting.chart">line</option>
```

事前定義されたトークンを使って、時間ウィンドウのもっとも早い時間ともっとも遅い時間、およびフィールドに対するその時間ウィンドウ内のもっとも早い/もっとも遅い値を取得します。

例：

```
<selection>
  <set token="selection.earliest">$start$</set>
  <set token="selection.latest">$end$</set>
  <set token="start.[fieldname]">$start.[fieldname]$</set>
  <set token="end.[fieldname]">$end.[fieldname]$</set>
</selection>
```

ドリルダウンリンクの設定にも使用できます。

```
<selection>
  <link>
```

属性

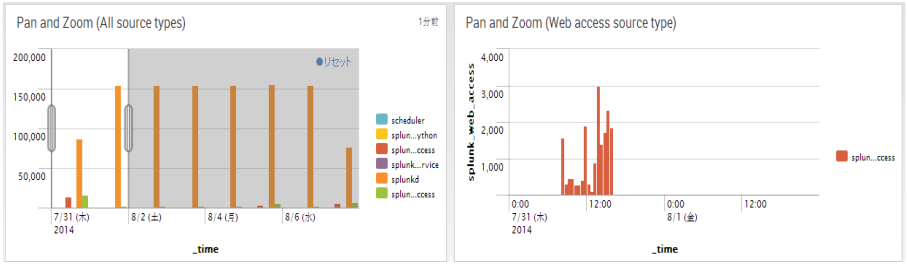
このエレメントの属性はありません。

例

左側のグラフで選択を行うと、右側のグラフがズームインされて、選択した領域の詳細が表示されます。

```
<dashboard>
  <label>Pan and Zoom</label>
  <row>
    <panel>
      <chart>
        <title>Pan and Zoom (All source types)</title>
        <search>
          <query>
            index=_internal  |  timechart count by sourcetype
          </query>
          <earliest>-7d@h</earliest>
          <latest>now</latest>
       </search>
       <option name="charting.axisX.scale">linear</option>
        <option name="charting.axisY.scale">log</option>
        <option name="charting.chart">line</option>
        <selection>
          <set token="selection.earliest">$start$</set>
          <set token="selection.latest">$end$</set>
          <set token="start.splunk_web_access">$start.splunk_web_access$</set>
          <set token="end.splunk_web_access">$end.splunk_web_access$</set>
        </selection>
        <option name="charting.axisTitleX.text">Last 7 Days</option>
      </chart>
    </panel>
    <panel>
      <chart>
        <title>Pan and Zoom (Web access source type)</title>
        <search>
          <query>
            index=_internal sourcetype=splunk_web_access
            |  timechart count by sourcetype
          </query>
        <earliest>$selection.earliest$</earliest>
        <latest>$selection.latest$</latest>
        </search>
        <option name="charting.chart">column</option>
        <option name="charting.legend.placement">none</option>
        <option name="charting.legend.masterLegend">null</option>
        <option name="charting.axisX.scale">linear</option>
        <option name="charting.axisY.scale">log</option>
        <option name="charting.axisTitleX.text">Selected Time Range</option>
      </chart>
    </panel>
  </row>
  <row>
    <panel>
      <html>
        <h3>Token values for the splunk_web_access selection</h3>
        <table border="0" cellpadding="12" cellspacing="0">
          <tr>
            <td>
              <p><b>Time range (epoch time)</b></p>
              <p>
                <b>$$selection.earliest$$</b>: $selection.earliest$<br/>
                <b>$$selection.latest$$</b>: $selection.latest$
              </p>
            </td>
            <td>
              <p><b>Count at the begining and end of time range.</b></p>
              <p>
                <b>$$start.splunk_web_access$$</b>: $start.splunk_web_access$<br/>
                <b>$$end.splunk_web_access$$</b>: $end.splunk_web_access$</p>
            </td>
          </tr>
        </table>
      </html>
    </panel>
  </row>
```

```
</dashboard>
```



ドリルダウンイベントトークン

動的ドリルダウンの場合、これらはイベントトークンとその値で、各視覚エフェクトタイプに利用できます。

- グラフイベントトークン
- イベントイベントトークン
- 地図イベントトークン
- 単一イベントトークン
- テーブルイベントトークン

グラフ (イベントトークン)

存在する場合、クリックされたフィールド名は、Y 軸のフィールドまたはシリーズの名前です (click.name2 と同様)。フィールドまたはシリーズの名前が利用できない場合、X 軸のフィールドまたはカテゴリが使用されます (click.name)。

データのプロパティ	説明
click.name	X 軸のフィールドまたはカテゴリの名前。凡例がクリックされた場合は利用できません。
click.value	X 軸のフィールドまたはカテゴリの値。凡例がクリックされた場合は利用できません。
click.name2	Y 軸のフィールドまたはシリーズの名前。
click.value2	Y 軸のフィールドまたはシリーズの値。凡例がクリックされた場合は利用できません。
row.<fieldname>	X 軸でクリックされた地点と同じ場所の Y 軸の任意のフィールド値。凡例がクリックされた場合は利用できません。
row.<x-axis-name>	X 軸の値。凡例がクリックされた場合は利用できません。
earliest/latest	クリックされたグラフセグメントの時間範囲、または存在しない場合は、サーチの時間範囲。

イベント (イベントトークン)

`click.name` の値は、後述するようにクリックのコンテキストによって異なります。

データのプロパティ	説明
click.name	クリックに関連するフィールド名。 フィールド名が曖昧なイベントビューアの場合： • raw イベント内の用語をクリックします。フィールド名として、`_raw` を設定します。 • イベントのタイムスタンプをクリックします。フィールド名として、`_time` を設定します。 • タグをクリックします。タグ名に従って、フィールド名を次のように指定します： tag::<field> (たとえば、host にタグを設定する場合は tag::host)

click.value	クリックに関連する値。
click.name2	`click.name` と同じです。
click.value2	`click.value` と同じです。
row.<fieldname>	各フィールド値を行として表示します。<fieldname>.
earliest/latest	次の、クリックされたイベントの時間範囲： earliest： `_time` latest： `(_time + 1 second)`

地図 (イベントトークン)

動的ドリルダウンの <condition> タグのフィールドは、常に click.name に対応しています。

データのプロパティ	説明
click.name	マーカーを表示する最初の、または唯一のフィールドの名前。
click.value	マーカーを表示する最初の、または唯一のフィールドの値。
click.name2	click.name と同じです。
click.value2	click.value と同じです。
click.lat.name	マーカーの場所を決定する緯度フィールドの名前。
click.lat.value	マーカーの Geo 場所の緯度値。
click.lon.name	マーカーの場所を決定する経度フィールドの名前。
click.lon.value	マーカーの Geo 場所の経度値。
click.bounds.<orientation>	マーカーが表すすべてのクラスタ化された場所の外部境界。 Orientation：south、west、north、east
row.<fieldname>	クリックされたマーカーの各フィールド値が、このフォームに表示されます。
earliest/latest	地図視覚エフェクトを表示するサーチの時間範囲。

単一 (イベントトークン)

動的ドリルダウンの <condition> タグのフィールドは、常に click.name に対応しています。

データのプロパティ	説明
click.name	単一値視覚エフェクトが表示するフィールド名。
click.value	単一値視覚エフェクトが表示するフィールド名。
click.name2	click.name と同じです。
click.value2	click.value と同じです。
row.<fieldname>	単一値が取得された同じ結果行の各フィールドを表示します。
earliest/latest	単一値視覚エフェクトを表示するサーチの時間範囲。

テーブル (イベントトークン)

動的ドリルダウンの <condition> タグのフィールドは、常に click.name2 に対応しています。

データのプロパティ	説明
click.name	テーブルに表示されている一番左のフィールドの名前。存在する場合は、常に `_time` になります。
click.value	クリックされた行の一番左側の列の値。
click.name2	クリックされた列名。
click.value2	クリックされた列の値。
row.<fieldname>	表示されていないフィールドも含めて、クリックされた行のすべてのフィールド値。
earliest/latest	クリックされたテーブル行の時間範囲、または存在しない場合は、サーチの時間範囲。

Link、Set、Unset

link

<link>

ドリルダウンまたは選択された入力項目のリンク先を指定します。

<link> は、<change>、<drilldown>、または <condition> の子タグにすることができます。

特定のフィールドまたは入力に対して一意のドリルダウン・アクションを設定する場合は、<link> を <condition> の子タグとして使用します。それ以外の場合は、<link> を <change> または <drilldown> の子タグとして使用します。

後述するように、相対パスまたは URL を使ってドリルダウンの宛先を指定する、さまざまな方法があります。

親エレメント

<drilldown><condition>
<change><condition>

```
<drilldown>
  <link>

<drilldown>
  <condition>
    <link>

<change>
  <link>

<change>
  <condition>
    <link>
```

属性

名前	タイプ	デフォルト	説明
field	フィールド名		**非推奨。次を使用：<condition field="[field]"...>** (<drilldown> のみ) 指定した列または行から取得する、テーブル内の値を指定します。`series` 属性と一緒に指定することはできません。 field 属性はサポートされていますが、<condition> タグでフィールドを指定することをお勧めします。
series	シリーズ名		**非推奨。次を使用：<condition field="[field]"...>** (<drilldown> のみ) 指定したシリーズから取得する、グラフ内の値を指定します。`field` 属性と一緒に指定することはできません。 series 属性はサポートされていますが、<condition> タグでシリーズを指定することをお勧めします。
target	テキスト	—	`<a>` HTTP タグの `target` 属性に対応しています。<link> エレメントのターゲットの指定は、<drilldown> エレメントに指定されたターゲットの値に優先します。 ドリルダウンを新しいウィンドウで表示するには、「_blank」を指定します。 ドリルダウンを同じウィンドウで表示するには、「_self」を指定します。 ドリルダウンを新しいウィンドウで表示するには、任意の文字列を指定します。このターゲットへのそれ以降の参照は、このウィンドウ内に開かれます。

親エレメント

<drilldown><condition>

```
1) <link> [viewname] </link>
```

```
2) <link> [path/viewname] </link>
3) <link> [path/viewname?form.token=$dest_value$] </link>
4) <link> [path/viewname?form.token=$dest_value$&earliest=$earliest$&latest=$latest$] </link>
5) <link> [URL?q=$dest_value$] </link>
```

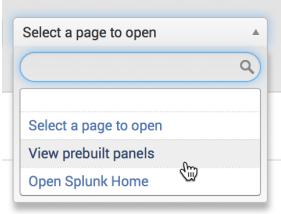
1. 指定したビューを使用します。これは、現在のダッシュボードと同じパスに存在する必要があります。
2. ダッシュボードに接続する相対パス。
3. フォームに接続する相対パスで、トークンを渡してフォームに記入します。
4. 元のサーチから、もっとも早い時間ともっとも遅い時間範囲を渡します。
   (特殊文字をエスケープ処理するために、CDATA を使用する必要があります。)
5. 宛先ページに値を渡す URL とクエリー引数。

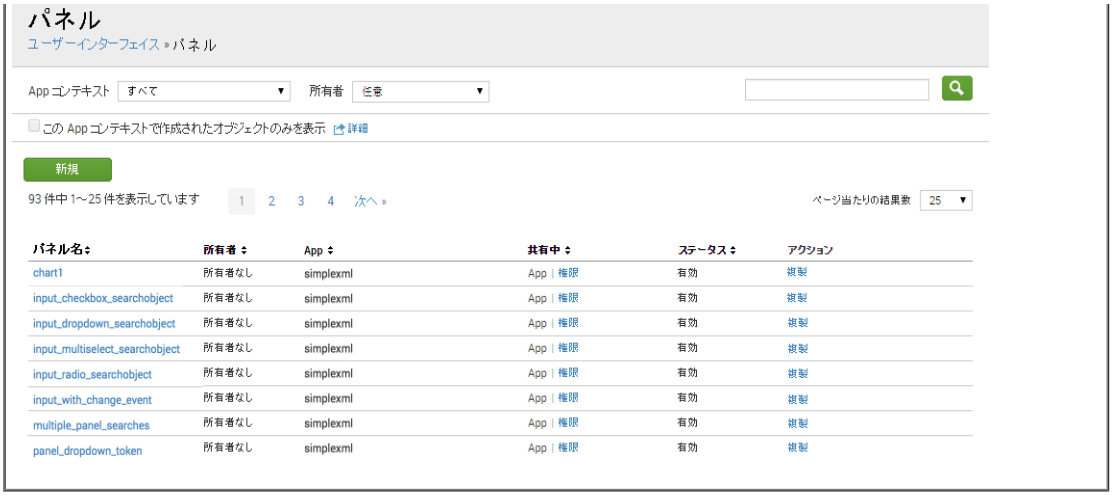
パス値	説明
path	現在のビューから宛先ビューへのパス。一般的に、パスは */app/App 名/* の形式で指定します。 ただし、ソースビューと宛先ビューの App コンテキストに基づいて、相対パスで指定することもできます。
viewname	宛先に使用する Splunk ビュー名。
$*dest_value*$	視覚エフェクトからの値の捕捉方法を指定します。各視覚エフェクトの詳細は、「ドリルダウンイベントトークン」を参照してください。
URL	Web ページへの URL を指定します。プロトコルも含めたフルアドレスを使用します。 例：`http://`。
q	URL を指定する場合、Web リソースへのクエリー文字列内の *`dest_value`* の値を指定するには q を使用します。

例

新しいページを開くには、条件付き入力で <link> を使用します。

```
<form>
. . .
<fieldset>
  <input type="dropdown" token="openNewPageToken">
    <label></label>
    <default>Select a page to open</default>
    <choice value="">Select a page to open</choice>
    <choice value="manager_page">View prebuilt panels</choice>
    <choice value="splk_page">Open Splunk home page</choice>
    <change>
      <condition value="manager_page">
        <link target="_blank">
           /manager/search/data/ui/panels?ns=-&pwnr=-&search=&count=25
        </link>
      </condition>
      <condition value="splk_page">
        <link target="_blank">
          http://splunk.com
        </link>
      </condition>
    </change>
  </input>
</fieldset>
. . .
</form>
```





set

<set>
ダッシュボード内の他のエレメントやサーチが利用できる、新たなグローバル・トークンを公開することができます。一般的には、フォーム入力の使用時、またはドリルダウンの使用時に、トークンを公開します。 フォーム入力の場合、特定の入力を取得するために、アクションのトークンを指定します。 ドリルダウンの場合、クリックされた時に取得する値を指定します。トークンを使って、値を動的に設定することができます。 フォーム入力の場合、<set> は、<change> または <condition> の子タグにすることができます。 ドリルダウンの場合、<set> は、<drilldown> または <condition> の子タグにすることができます。 ドリルダウン用の特定の入力またはフィールドに対して、一意のアクションを設定する場合は、<set> を <condition> の子タグとして使用します。または、すべての入力またはすべてのフィールドに対するアクションを指定するために、<set> を <change> または <drilldown> の子タグとして使用します。
親エレメント <change> <condition> <drilldown > <condition> <change> <drilldown>

トークンの値を設定するには、2 種類の方法があります。
 	
1.テンプレートを使って入力トークンと静的部を組み合わせて、新たなトークン値を形成します。テンプレートを利用して、値の設定時に複数のトークンを参照することができます。また、|s トークン・フィルタを使って、値の引用符を指定することもできます。
 	
`<set token="Token Name">sourcetype=$click.value\|s$</set>`
 	
2.prefix および suffix 属性を使って、入力トークンの静的部を指定します。以下は、上記のテンプレートの例と同等です。
 	
`<set token="Token Name" prefix="sourcetype=&quot;" suffix="&quot;">$click.value$</set>`

属性 **名前** / **タイプ** / **デフォルト** / **説明** **token** / トークン名 / / **必須** 同じページのターゲット視覚エフェクトが使用するトークン名。 **prefix** / テキスト / / トークンの値の前に配置する文字列。 **suffix** / テキスト / / トークンの値に追加する文字列。
例 テーブルをクリックすると、グラフ視覚エフェクトのサーチが使用するトークンが設定されます。

```
<dashboard>
  <label>In-page Drilldown</label>
  <row>
    <panel>
      <table>
        <title>Set sourcetype token on click</title>
        <search>
          <query>
          index=_internal | stats count by sourcetype
          </query>
        <earliest>-1h</earliest>
        <latest>now</latest>
        </search>
        <drilldown>
          <condition field="sourcetype">
            <set token="sourcetype">$click.value2$</set>
          </condition>
        </drilldown>
      </table>
      <chart>
        <title>Chart for $sourcetype$</title>
        <search>
          <query>
            index=_internal sourcetype=$sourcetype$ | timechart count by sourcetype
          </query>
          <earliest>-1h</earliest>
          <latest>now</latest>
        </search>
      </chart>
    </panel>
  </row>
</dashboard>
```

unset

<unset>
<unset> を使って、前に設定されたトークンを削除します。
親エレメント <change> <condition> <drilldown > <condition> <change> <drilldown>

```
<unset token="Token Name">
```

属性

名前	タイプ	デフォルト	説明
token	トークン名		**必須** 以前に設定されているけれども、無視するトークン名。

例

<set> および <unset> を使って、使用する視覚エフェクトを定義します。

パネルを非表示にするには、トークン定義を使用します。

```
<dashboard>
  <label>Example for <set> and <unset></label>
  <row>
    <panel>
      <table>
        <title>Set sourcetype token</title>
```

```
        <search>
          <query>
          index=_internal | stats count by sourcetype
          </query>
        <earliest>-1h</earliest>
        <latest>now</latest>
        </search>
        <drilldown>
          <!-- For the sourcetype field clicked: -->
          <!-- Set token to display a chart -->
          <!-- Unset token to display a table -->
          <condition field="sourcetype">
            <set token="sourcetype">$row.sourcetype$</set>
            <set token="showChart">foo</set>
            <unset token="showTable"></unset>
          </condition>
          <!-- For any other field clicked: -->
          <!-- Set token to display a table -->
          <!-- Unset token to display a chart -->
          <condition field="*">
            <set token="sourcetype">$row.sourcetype$</set>
            <set token="showTable">foo</set>
            <unset token="showChart"></unset>
          </condition>
        </drilldown>
      </table>
    </panel>

    <!-- Hide the html panel when either token is present -->
    <!-- Click in the original table to set either token -->
    <panel>
      <html rejects="$showTable$, $showChart$">
        <h2>Details</h2>
          <div style="padding: 50px; margin: 0 auto; width: 350px;">
          <div class="alert alert-warning">
          <i class="icon-alert"/>
            Click on a row in the table on the left to show details.
          </div>
        </div>
      </html>
      <!-- if showChart token is set, display results here -->
      <chart depends="$showChart$">
        <title>Details for $submitted:sourcetype|s$</title>
        <search>
          <query>
          index=_internal sourcetype=$sourcetype|s$
          | timechart count by sourcetype
          </query>
          <earliest>-1h</earliest>
          <latest>now</latest>
        </search>
      </chart>
      <!-- if showCTable token is set, display results here -->
      <table depends="$showTable$">
        <title>Details for $submitted:sourcetype|s$</title>
        <search>
          <query>
          index=_internal sourcetype=$sourcetype|s$
          | timechart bins=10 count by sourcetype
          </query>
          <earliest>-1h</earliest>
          <latest>now</latest>
        </search>
        <option name="wrap">true</option>
        <option name="rowNumbers">false</option>
        <option name="dataOverlayMode">none</option>
        <option name="drilldown">cell</option>
        <option name="count">10</option>
      </table>
    </panel>
  </row>
</dashboard>
```

グラフ設定リファレンス

グラフの概要

<chart> エレメントは、自在に設定可能なパネル視覚エフェクトです。

<chart>
サーチ・データをグラフに表示するパネル。保存済みレポートには、グラフの書式設定パラメータが含まれています。保存済みサーチには含まれていません。詳細は、「レポートの保存と他のユーザーとの共有」を参照してください。 グラフパネルに保存済みレポートをロードする場合、保存済みレポートの書式設定もロードされます。ただし、グラフの書式設定はグラフオプションで上書きすることができます。 グラフは名前付きオプションを使って、グラフ固有のプロパティを指定します。このリファレンスには、グラフの設定可能なすべてのプロパティに関するセクションが含まれています。
親エレメント <row> <panel>
<chart> <title> (0..1) <search> (0..1) <earliest> (0..1) <latest> (0..1) <drilldown> (0..n) <selection> (0..n, for charts of type area, line, and column only) <option name="[property]"> (0..n)

全般的なグラフのプロパティ

ここでは、すべてのグラフに適用されるプロパティを記載しています。

プロパティ	タイプ	デフォルト	説明
charting.chart	(area | bar | bubble | column | fillerGauge | line | markerGauge | pie | radialGauge | scatter)	列	グラフタイプを設定します。
charting.data.count	数値	1000	取得する結果数。すべての結する場合は、0 を設定しま **警告：**すべての結果を取得0 を設定すると、パフォーマな影響が出る可能性があり
charting.data.fieldListMode	(show_hide | hide_show)	hide_show	fieldShowList および fieldH フィルタを適用する順番。
charting.data.fieldShowList	フィールドの配列	—	明示的に結果を表示するフィリスト。
charting.data.fieldHideList	フィールドの配列	—	明示的に結果から非表示にすルドのリスト。
charting.data.jobID	テキスト	—	サーチジョブ ID。
charting.data.preview	論理値	false	結果をプレビューするかどうます。
charting.data.offset	数値	0	最初に取得された結果のオフ
charting.data.search	サーチ文字列	—	必要に応じて結果に適用するサーチ。
charting.drilldown	(all | none)	all	all：ドリルダウン有効。

charting.drilldown	(all | none)	all	**none**：ドリルダウン無効。

charting.fieldColors	16 進色のマップ。
	
	説明を参照。
		—	各フィールドに対して使用す
			値のマップです。
			
			マップは波括弧で囲んだ、キ
			アのカンマ区切りリストです
			
			キーと値はコロンで区切りま
			
			例：
			
			`{key1:value1,key2:value2,â€¦,`
			
			キーまたは文字列値内の以下
			字は、二重引用符でエスケー
			ます。
			
			`[]{}(),:"`
			
			既存の二重引用符または円記
			スラッシュ) をエスケープ処
			は、その前に円記号 (バック
			シュ) を指定します。
			
			例については、「グラフ内の
			ドへのカスタム色の指定」を
			ください。

charting.legend.labels	ラベルの CSV	—	凡例を事前設定するための、 リスト。
charting.legend.labelStyle.overflowMode	(ellipsisEnd | ellipsisMiddle | ellipsisNone | ellipsisStart)	ellipsisMiddle	レイアウト境界をオーバーフ ラベルの、省略されたテキス 記号 (...) で表示する方法を決 す。 **ellipsisStart**：先頭のテキ します。 **ellipsisMiddle**：行の中央 を省略します。 **ellipsisEnd**：レイアウト境 ストを省略します。 **ellipsisNone**：テキストの 的に無効にします。
charting.legend.masterLegend	N/A		属性が存在している場合、ダ ボード内の他のパネルとの凡 期を無効にします。 **注意**：唯一有効な値は、空の す。値を指定した場合、属性 れます。
charting.seriesColors	16 進色のマップ。	以下を参照してください。*	16 進値の配列を使って、グ リーズの色を定義します。 **注意**：特定のフィールドに割 適用するには、**charting.fi** プロパティを使用します。
drilldown	(all | none)	—	**非推奨。** charting.drilldown ます。
height	数値	—	グラフの高さ (ピクセル)。

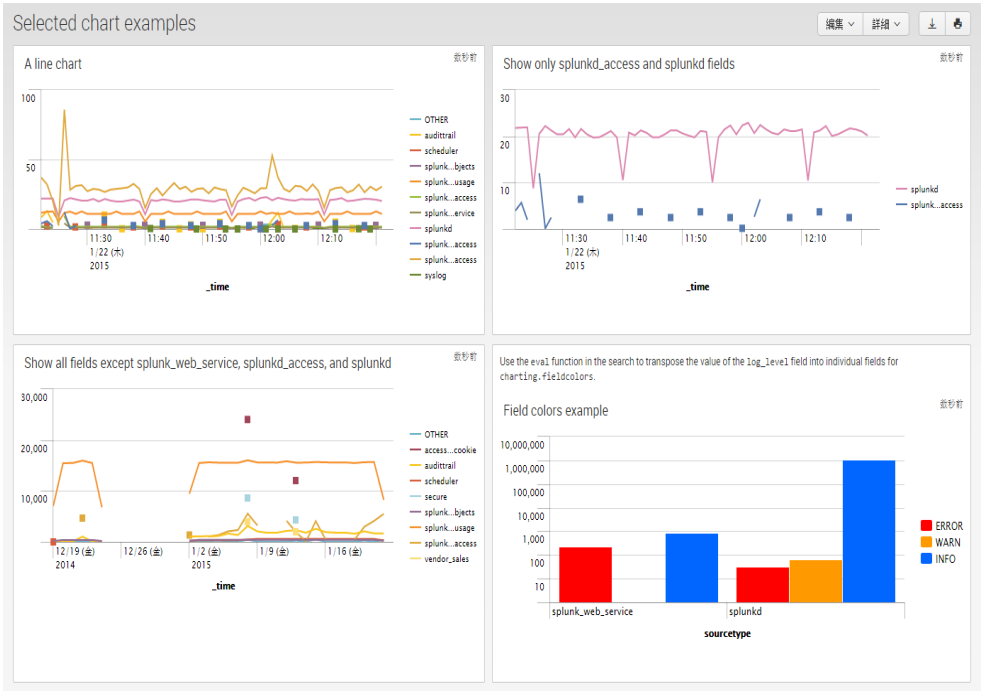
*charting.seriesColors のデフォルト値：

```
[0x6CB8CA, 0xFAC61D, 0xD85E3D, 0x956E96, 0xF7912C, 0x9AC23C, 0x998C55, 0xDD87B0, 0x5479AF, 0xE0A93B,
0x6B8930, 0xA04558, 0xA7D4DF, 0xFCDD77, 0xE89E8B, 0xBFA8C0, 0xFABD80, 0xC2DA8A, 0xC2BA99, 0xEBB7D0,
0x98AFCF, 0xECCB89, 0xA6B883, 0xC68F9B, 0x416E79, 0x967711, 0x823825, 0x59425A, 0x94571A, 0x5C7424,
0x5C5433, 0x85516A, 0x324969, 0x866523, 0x40521D, 0x602935]
```

全般的なグラフプロパティ：選択された例

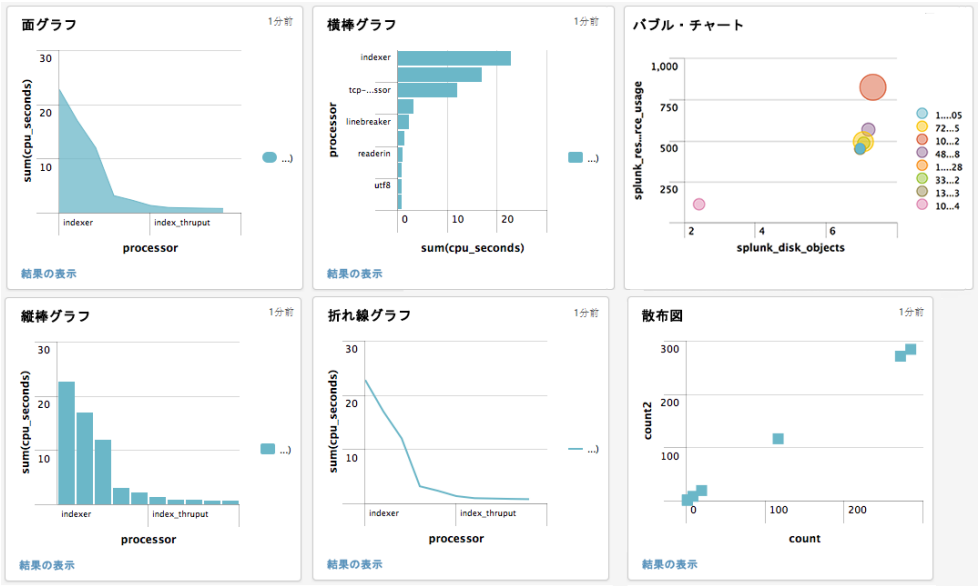
```
<dashboard>
  <label>Selected chart examples</label>
  <row>
    <panel>
      <chart>
        <title>A line chart</title>
        <search>
          <query>
            index=_internal source="*metrics.log"
            group=per_sourcetype_thruput
            | timechart sum(kb) by series
          </query>
          <earliest>-1h</earliest>
          <latest>now</latest>
        </search>
        <option name="charting.chart">line</option>
      </chart>
    </panel>

    <panel>
      <chart>
        <title>Show only splunkd_access and splunkd fields</title>
        <search>
          <query>
            index=_internal source="*metrics.log"
            group=per_sourcetype_thruput
            | timechart sum(kb) by series
          </query>
          <earliest>-1h</earliest>
          <latest>now</latest>
        </search>
        <option name="charting.data.fieldShowList">
          ["splunkd_access", "splunkd"]
        </option>
        <option name="charting.chart">line</option>
      </chart>
    </panel>
  </row>

  <row>
    <panel>
      <chart>
        <title>Show all fields except splunk_web_service, splunkd_access, and splunkd</title>
        <search>
          <query>
            index=_internal source="*metrics.log"
            group=per_sourcetype_thruput
            | timechart sum(kb) by series
          <earliest>-1h</earliest>
          <latest>now</latest>
          </query>
        </search>
        <option name="charting.data.fieldHideList">
          ["splunk_web_service", "splunkd_access", "splunkd"]
        </option>
        <option name="charting.chart">line</option>
      </chart>
    </panel>

    <panel>
      <html>
        Use the <tt>eval</tt> function in the search to transpose
        the value of the <tt>log_level</tt> field into individual
        fields for <tt>charting.fieldcolors</tt>.
      </html>
      <chart>
        <title>Field colors example</title>
        <search>
          <query>
            index = _internal log_level=* | stats
            count(eval(log_level="ERROR")) as ERROR
```

```
            count(eval(log_level="WARN")) as WARN
            count(eval(log_level="INFO")) as INFO
            by sourcetype
          </query>
          <earliest>-7d@h</earliest>
          <latest>now</latest>
        </search>
        <option name="charting.axisY.scale">log</option>
        <option name="charting.chart">column</option>
        <option name="charting.fieldColors">
          {"ERROR": 0xFF0000, "WARN": 0xFF9900, "INFO":0x0066FF, "NULL":0xC4C4C0}
        </option>
        <option name="charting.legend.placement">right</option>
      </chart>
    </panel>
  </row>
</dashboard>
```



面、バブル、横棒、縦棒、折れ線グラフ、および散布図

面、横棒、縦棒、および折れ線グラフ固有のプロパティで、すべてに X 軸と Y 軸が含まれています。



プロパティ	タイプ	デフォルト	説明
charting.axisLabelsX.axisVisibility charting.axisLabelsY.axisVisibility	(show | hide)	表示	軸の線を表示するかどうかを示します。
charting.axisLabelsX.extendsAxisRange charting.axisLabelsY.extendsAxisRange	論理値	true	軸の範囲を、主軸目盛全体にスナップするように拡張するかどうかを示します。
charting.axisLabelsX.integerUnits charting.axisLabelsY.integerUnits	論理値	false	主軸単位を最寄りの整数に四捨五入するかどうかを示します。
charting.axisLabelsX.majorLabelStyle.overFlowMode	(ellipsisMiddle | ellipsisNone)	ellipsisNone	目盛りの間隔に軸ラベルを省略するかどうかを示します。
charting.axisLabelsX.majorLabelStyle.rotation	(-90 | -45 | 0 | 45 | 90)	0	軸ラベルの回転 (角度)。 時計回りに回転する場合は正の値。負の値は反時計回り。
			主軸目盛ラベルの表示を制御します。 大量の結果が表示されるような場合でも常にラベルを表示する場合は、show を設定します。

charting.axisLabelsX.integerUnits charting.axisLabelsY.integerUnits	(auto | show | hide)	auto	**auto**；利用可能なスペースが重複することなく、理解しやすさを維持するために、個別の主要ラベルを表示/非表示にします。 **show**：重複が発生した場合でも、すべての主要ラベルを表示します。 **hide**：すべての主要ラベルを非表示にします。
charting.axisLabelsX.majorTickSize charting.axisLabelsY.majorTickSize	数値	6	主軸目盛のサイズ (ピクセル)。
charting.axisLabelsX.minorTickSize charting.axisLabelsY.minorTickSize	数値	6	副軸目盛のサイズ (ピクセル)。
charting.axisLabelsX.majorTickVisibility charting.axisLabelsY.majorTickVisibility	(auto | show | hide)	auto	主軸目盛りを表示するかどうかを指定します。 **auto**；対応するラベルが表示される場合にのみ、主軸目盛りを表示します。 **show**：ラベルの表示有無に関係なく、すべての主軸目盛りを表示します。 **hide**：すべての主軸目盛りを非表示にします。
charting.axisLabelsX.majorUnit charting.axisLabelsY.majorUnit	(Number | auto)	auto	数値軸に沿って主軸目盛を配置する間隔の単位です。 デフォルトでは、この値は関連する軸のスケールに基づいて自動的に算出されます。
			副軸目盛りを表示するかどうかを指定しま

			す。 **auto**：対応するラベルが表示される場合にのみ、副軸目盛りを表示します。 **show**：ラベルの表示有無に関係なく、すべての副軸目盛りを表示します。 **hide**：すべての副軸目盛りを非表示にします。
charting.axisLabelsX.minorTickVisibility charting.axisLabelsY.minorTickVisibility	(auto | show | hide)	auto	
charting.axisX.includeZero charting.axisY.includeZero	論理値	false	軸範囲に 0 を含めるかどうかを示します。
charting.axisX.maximumNumber charting.axisY.maximumNumber	数値	auto	軸範囲の最大値を設定します。
charting.axisX.minimumNumber charting.axisY.minimumNumber	数値	auto	軸範囲の最小値を設定します。
charting.axisX.scale charting.axisY.scale	(linear | log)	線形	線形または対数スケールを使用します。 X 軸の対数スケールは、バブル・チャートおよび散布図のみがサポートしています。
charting.axisTitleX.text charting.axisTitleY.text	テキスト	—	X 軸または Y 軸のタイトルを指定します。
charting.axisTitleX.visibility charting.axisTitleY.visibility	(visible | collapsed)	visible	X 軸または Y 軸のタイトルを表示するかどうかを指定します。
charting.chart.resultTruncationLimit	数値	状況によって異なります。説明を参照。	グラフに表示するオブジェクト数合計の、デフォルトの表示制限に優先します。これらはデカルト (2 軸) グラフに適用されます。 使用するグラフのタイプによって、デフォルト値は異なります。例：

			• 縦棒：1200 • 横棒：1200 • 折れ線：2000 • 面：2000
charting.gridlinesX.showMajorLInes charting.gridlinesY.showMajorLInes	論理値	True	主軸グリッド線を表示するかどうかを示します。
charting.gridlinesY2.showMajorLInes	論理値	False	主軸グリッド線を表示するかどうかを示します。
charting.gridlinesX.showMinorLInes charting.gridlinesY.showMinorLInes charting.gridlinesY2.showMinorLInes	論理値	True	副軸グリッド線を表示するかどうかを示します。
charting.layout.splitSeries	論理値	False	複数シリーズグラフを個別のグラフに分割して、それをシリーズごとに上から下にスタックします。
charting.legend.placement	(top | left | bottom | right | none)	右	ラベルの配置場所。

面グラフのプロパティ

プロパティ	タイプ	デフォルト	説明
charting.areaFillOpacity	0～1.0	.75	面グラフの不透明度を設定します。 1.0 は面グラフが単色であることを表しています。0 は面グラフが透明であることを表しています。
charting.chart.nullValueMode	(gaps | zero | connect)	ギャップ	null 値の処理方法を指定します。
charting.chart.showLines	論理値	true	面グラフに線を表示するかどうかを示します。
charting.chart.stackMode	(default | stacked | stacked100)	default	スタック面グラフを設定します。

横棒グラフのプロパティ

プロパティ	タイプ	デフォルト	説明
charting.chart.barSpacing	数値	1	横棒グラフのバー間のスペースを指定します (ピクセル)。
charting.chart.seriesSpacing	数値	—	横棒グラフのクラスタ化されたシリーズ間のスペースを指定します (ピクセル)。
charting.chart.stackMode	(default | stacked | stacked100)	default	スタック横棒グラフを設定します。

バブルグラフのプロパティ

プロパティ	タイプ	デフォ	説明

		ルト	
charting.chart.bubbleMaximumSize	数値	50	各バブルの最大サイズをピクセルで指定します。
charting.chart.bubblesMinimumSize	数値	10	各バブルの最低サイズをピクセルで指定します。
charting.chart.bubbleSizeBy	(area | diameter)	area	バブルのサイズを決定するのが、面積または直径であることを指定します。

縦棒グラフのプロパティ

プロパティ	タイプ	デフォルト	説明
charting.chart.columnSpacing	数値	1	縦棒間の間隔を指定します (ピクセル)。
charting.chart.seriesSpacing	数値	—	横棒グラフのクラスタ化されたシリーズ間のスペースを指定します (ピクセル)。
charting.chart.stackMode	(default | stacked | stacked100)	default	スタック縦棒グラフを設定します。

折れ線グラフのプロパティ

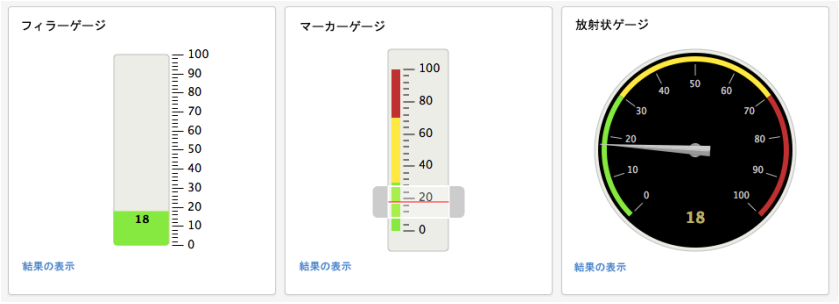
プロパティ	タイプ	デフォルト	説明
charting.chart.nullValueMode	(gaps | zero | connect)	ギャップ	null 値の処理方法を指定します。
charting.chart.showMarkers	論理値	true	折れ線グラフにマーカーを描画するかどうかを示します。
charting.chart.stackMode	(default | stacked | stacked100)	default	スタック折れ線グラフを設定します。
charting.lineDashStyle	(dashDot | dot | longDash | longDashDot | longDashDotDot | shortDash | shortDot | shortDashDot | solid)	solid	グラフ内のすべてのインライン・サーチのダッシュのスタイルを示します。

散布図のプロパティ

プロパティ	タイプ	デフォルト	説明
charting.chart.markerSize	数値	4	マーカーのサイズを示します (ピクセル)。

ゲージグラフ

ゲージグラフ固有のプロパティ：



プロパティ	タイプ	デフォルト	説明
charting.gaugeColors	[16 進,...]	[0x84E900, 0xFFE800,	16 進数の色値配列。ここから範囲バンドの色が生成されます。 色は、配列に示されている順序で表示されます。 たとえば、gaugeColors を次の値に変更することで、デフォルトの緑-黄-赤の色順序を反転することができます。

		0xBF3030]	`[0xBF3030,0xFFE800,0x84E900]` 任意の数の色を指定できます。ゲージの範囲が rangeColors よりも多いまたは少ない場合、Splunk は必要に応じて色を補間します。これは、サーチ言語の範囲間隔または rangeValues パラメータの指定によって行われます。
charting.chart.majorUnit	数値	auto	主軸目盛の間隔を指定します (ピクセル)。
charting.chart.rangeValues	数値配列。	—	ゲージが示す全体的な数値範囲を表す数値配列、および範囲全体内の色分けされたサブ範囲の相対サイズ。 たとえば、次の範囲は `[0,30,70,100]` ゲージが 0 から始まり、100 で終了することを示します。また、個別のフィラー色で示される 3 つのサブ範囲に分かれています。サーチが値 71 を返した場合、ゲージのフィラーはその値まで上昇し、上の範囲 (71～100) に割り当てられている色で表示されます。 **注意**：シンプル XML で範囲を指定した場合、その設定はダッシュボードパネルがベースにしているサーチに指定されている範囲値に優先します。
charting.chart.showLabels	論理値	True	ラベルを表示するかどうかを示します。
charting.chart.showMajorTicks	論理値	True	主軸目盛を表示するかどうかを示します。
charting.chart.showMinorTicks	論理値	False	副軸目盛を表示するかどうかを示します。
charting.chart.showValue	論理値	True	ゲージに値を表示するかどうかを示します。
charting.chart.style	(minimal | shiny)	shiny	ゲージのスタイルを指定します。 **shiny**：クロム、シェード、その他の機能などの、現実世界のゲージを模倣した、グラフィック的にスタイル化されたゲージ。 **minimal**：ゲージの基本版です。
charting.chart.usePercentageRange	論理値	False	範囲値をパーセントで書式設定するかどうかを示します。
charting.chart.usePercentageValue	論理値	False	ゲージ値をパーセントで書式設定するかどうかを示します。

フィラーゲージ固有のプロパティ

プロパティ	タイプ	デフォルト	説明
charting.chart.orientation	(x | y)	y	ゲージの向きを設定します。 **x**；horizontal **y**：vertical

マーカーゲージ固有のプロパティ

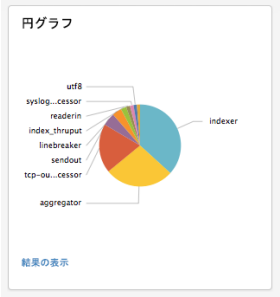
プロパティ	タイプ	デフォルト	説明
charting.chart.orientation	(x | y)	y	ゲージの向きを設定します。 **x**；horizontal **y**：vertical

放射状ゲージ固有のプロパティ

プロパティ	タイプ	デフォルト	説明
charting.chart.rangeArcAngle	数値	270	範囲の円弧の長さ (度)。 正の値は時計回り。負の値は反時計回り。
charting.chart.rangeStartAngle	数値	45	円弧の範囲を描画し始める角度。円弧は時計回りで、ゲージの下から開始します。

円グラフ

円グラフ固有のプロパティ：



プロパティ	タイプ	デフォルト	説明
charting.chart.sliceCollapsingLabel	テキスト	その他	まとめたスライスに付けるラベル。
charting.chart.sliceCollapsingThreshold	数値	0.01	小さなスライスを統合スライスにまとめるための閾値。 有効な値は 0～1 です。 0 の場合、まとめることはありません。1 の場合、すべてのスライスが単一の円グラフにまとめられます。 デフォルト値の 0.01 の場合、円全体の 1% 未満のスライスがまとめられます。
charting.chart.showLabels	論理値	true	ラベルを表示するかどうかを示します。
charting.chart.showPercent	論理値	false	パーセント値をラベルで表示するかどうかを示します。

トークン参照

トークンは、シンプル XML ダッシュボード内で値を渡すために利用できる変数です。このリストには、さまざまな状況下で利用できる各種トークンが記載されています。各エントリには簡単な説明と、詳細情報と例に関するリンクが記載されています。

トークン使用の概要については、「ダッシュボードでのトークンの使用」を参照してください。

トークン・タイプ	エレメント	説明
フォーム入力	<input>	入力から選択された値を参照するための、ユーザー定義入力。 「フォーム入力のトークンの定義」を参照してください。 フォーム入力の例

タイム・ピッカー入力	<input type="time">	複数のタイム・ピッカーをダッシュボード内の複数のパネルと関連付ける、オプションのユーザー定義入力トークン。 時間範囲を取得するために、`earliest` および `latest` 修飾子を含んでいます。 「時間入力のトークンの定義」を参照してください。 時間入力の例
ドリルダウン・イベント	<drilldown>	グラフ内のクリックから値を取得する事前定義トークン。動的ドリルダウン操作は、ドリルダウン・ターゲットにアクセスする際に、ソース・グラフから取得された値を使用します。 事前定義トークンの一覧については、「ドリルダウン・イベント・トークン」を参照してください。 「ドリルダウンのトークンの定義」を参照してください。 動的ドリルダウンの例
パン/ズーム・イベント	<selection>	パン/ズーム操作の値の範囲を取得する事前定義トークン。トークン値は、グラフ上のユーザー選択に適用されます。トークンのコンテキストはグラフ専用です。ダッシュボード内の値にアクセスするために、ユーザー定義トークンにトークン値をコピーします。 `start` と `end` は、グラフの X 軸の選択領域の開始/終了値を取得します。たとえば、時間グラフの選択範囲の開始/終了時刻を取得します。 `start.<field>`と `end.field` は、グラフの Y 軸の選択領域の開始/終了値を取得します。たとえば、時間グラフで選択を行うと、<field> で示されるシリーズのイベント数が取得されます。 「パン/ズーム・グラフ・コントロール用トークンの定義」を参照してください。 「パン/ズーム・グラフ・コントロール用トークンの定義」には、時間グラフを使った例が記載されています。
条件付きドリルダウン・アクション	<drilldown> <condition> <link> <set>|<unset>	条件付き操作を設定するための、condition エレメント内のユーザー定義トークン。条件による操作には、以下の事項が含まれています。 • 条件に基づいてトークン値を設定する。 • 視覚エフェクトの複数値フィールド用の値を選択する。 • トークン値に基づいて、開くビューを選択する。 • 条件に基づいて、パネルを非表示/表示する。 「<drilldown> エレメントによる条件付き操作用トークンの定義」を参照してください。 複数値フィールドへのダッシュボード・リンクの例
条件付きフォーム入力アクション	<input> <change> <condition> <link> <set>|<unset>	トークンの条件値に基づいてサーチを変更する、または表示する視覚エフェクトを選択する、condition エレメント内のユーザー定義トークン。 「フォーム入力による条件付き操作用トークンの定義」を参照してください。 フォーム入力を使った条件操作の例
宛先アクションの設定	<input>|<drilldown> <condition> <link> <set>|<unset>	開くターゲット・ページを指定するために、トークンを設定/設定解除します。 <input> エレメントまたは <drilldown> エレメントと併用することができます。<condition> エレメントは、アクションの条件を定義します。<link> エレメントはトークンを使用して、ターゲットを開きます。 「動的ドリルダウンの基本」を参照してください。 フォームにリンクするダッシュボードの例